CME Group

CME Legacy FIX/FAST Market Data

Functional Specification

12/2/2015

Futures trading is not suitable for all investors, and involves the risk of loss. Futures are a leveraged investment, and because only a percentage of a contract's value is required to trade, it is possible to lose more than the amount of money deposited for a futures position. Therefore, traders should only use funds that they can afford to lose without affecting their lifestyles. And only a portion of those funds should be devoted to any one trade because they cannot expect to profit on every trade.

All references to options refer to options on futures.

CME Group is a trademark of CME Group Inc. The Globe Logo, CME, Chicago Mercantile Exchange, Globex, iLink, E-mini, CME EOS Trader, Galax-C, FirmSoft, CME DataSuite, and CME DataMine are trademarks of Chicago Mercantile Exchange Inc. New York Mercantile Exchange, NYMEX, miNY, and ClearPort are registered trademarks of the New York Mercantile Exchange, Inc. COMEX is a trademark of Commodity Exchange, Inc.

FIX™ and FAST™ are trademarks of FIX Protocol Limited. FIX/FAST℠ is a service mark of FIX Protocol Limited.

Dow Jones℠, Dow Jones AIG Commodity Index℠, The Dow℠, Dow Jones Industrial Average℠, and DJIA℠ are service marks of Dow Jones & Company, Inc. and American International Group, Inc. (AIG) and have been licensed for use for certain purposes by the Board of Trade of the City of Chicago, Inc (CBOT®). CBOT futures and options on futures contracts based on the Dow Jones Industrial Average℠ are not sponsored, endorsed, sold or promoted by Dow Jones℠, and Dow Jones℠ makes no representation regarding the advisability of trading such product(s).

BM&FBOVESPA™ is a trademark of BM&FBOVESPA, KRX™ is a trademark of Korea Exchange, DME™ is a trademark of Dubai Mercantile Exchange, BMD™ is a trademark of Bursa Malaysia, BMV™ is a trademark of Bolsa Mexicana De Valores.

All other trademarks are the property of their respective owners.

The information within this document has been compiled by CME Group for general purposes only. CME Group assumes no responsibility for any errors or omissions. Additionally, all examples in this brochure are hypothetical situations, used for explanation purposes only, and should not be considered investment advice or the results of actual market experience.

All matters pertaining to rules and specifications herein are made subject to and are superseded by official CME, CBOT, and NYMEX rules. Current rules should be consulted in all cases concerning contract specifications.

Copyright © 2011 CME Group Inc. All rights reserved.

Legacy FIX FAST

- CME Globex Legacy Market Data
- CME Globex Legacy Market Data to MDP 3.0 Market Data Conversion

CME Globex Legacy Market Data

THIS SPECIFICATION IS FOR THE LEGACY FIX/FAST MARKET DATA MESSAGING FORMAT ONLY.

Use this search bar to search topics within CME Globex Legacy Market Data.

The selected root page could not be found.

CME Globex market data is transmitted on a given Market Data Platform (MDP) channel in the FIX message format using FAST compression.

This topic describes how client systems connect to and process messages sent over MDP.

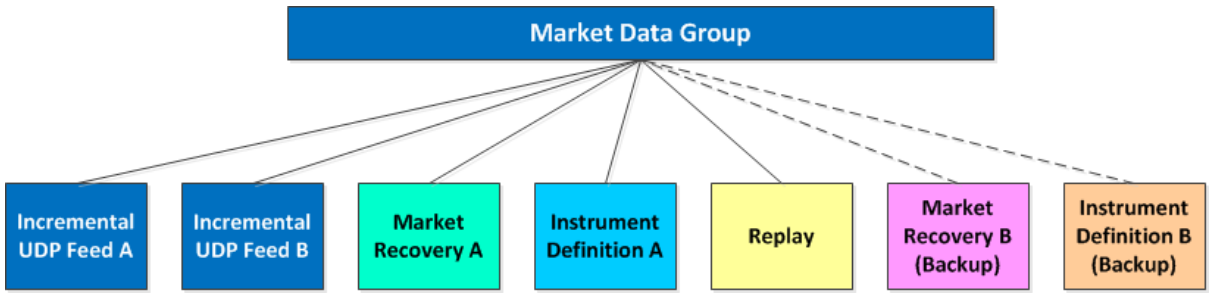
- Market Data Dissemination
- Market Data - Incremental Book Management
- Market Data - Real Time Statistics and Market Behavior Events
- Market Data - Recovery
- Legacy Market Data - Message Specification

Market Data Dissemination

The CME Group Market Data Platform (MDP) disseminates real-time bid, ask, trade, and statistical data for CME Group markets and also provides recovery and supporting services for market data processing.

Market Data Group

Market data is organized by Market Data Group, which is a configuration of MDP channels providing all market data required to support markets for a given product or set of products.



Incremental Feed

UDP Feed A and UDP Feed B are used to disseminate CME Group incremental market data using bandwidth efficient FAST encrypted FIX messages. All FIX message types (e.g., book update, statistics, quotes, instrument definitions, trades, instrument / instrument group, security definitions, CME Globex status) are sent through both UDP Feed A and UDP Feed B applicable Market Data Groups. This duality minimizes the chance of message loss due to UDP. Each FIX message is sent down both feeds. Please note that a single FIX/FAST Market Data Incremental Refresh (tag 35-MsgType=X) message can contain multiple updates for multiple instruments.

UDP Feed A and UDP Feed B should be used for arbitration.

Market Recovery Feed

Market Recovery (UDP) Feed A is used to disseminate CME Group market data snapshots for all books with any activity since the beginning of the week. Market Recovery (UDP) Feed B functions as a backup in the event that Feed A becomes inoperable. Market Recovery (UDP) Feed A and Market Recovery (UDP) Feed B should not be used for arbitration.

Book updates (outright and implied) and statistics are sent through the applicable incremental market data channel. A single FIX/FAST message contains the market state for a given instrument. Snapshots are replayed at a constant flow of configurable TPS. Expired instruments are included on the Market Recovery feed.

> CME strongly recommends that the Market Recovery feeds be used for recovery purposes only. Once client systems have retrieved recovery data, client systems should stop listening to the Market Recovery feeds.

Instrument Definition Feed

Instrument Definition (UDP) Feed A is used to disseminate CME Group instrument definitions. A single FIX/FAST message contains the definition of a given instrument. Instrument definitions are replayed at a constant flow of configurable TPS.

See Market Recovery and Instrument Definition Rates for additional information.

Expired instruments are not included on the Instrument Definition channel. Instrument Definition (UDP) Feed B functions as a backup in the event that Feed A becomes inoperable.

> Instrument Definition (UDP) Feed A and Instrument Definition (UDP) Feed B should not be used for arbitration.

TCP Replay

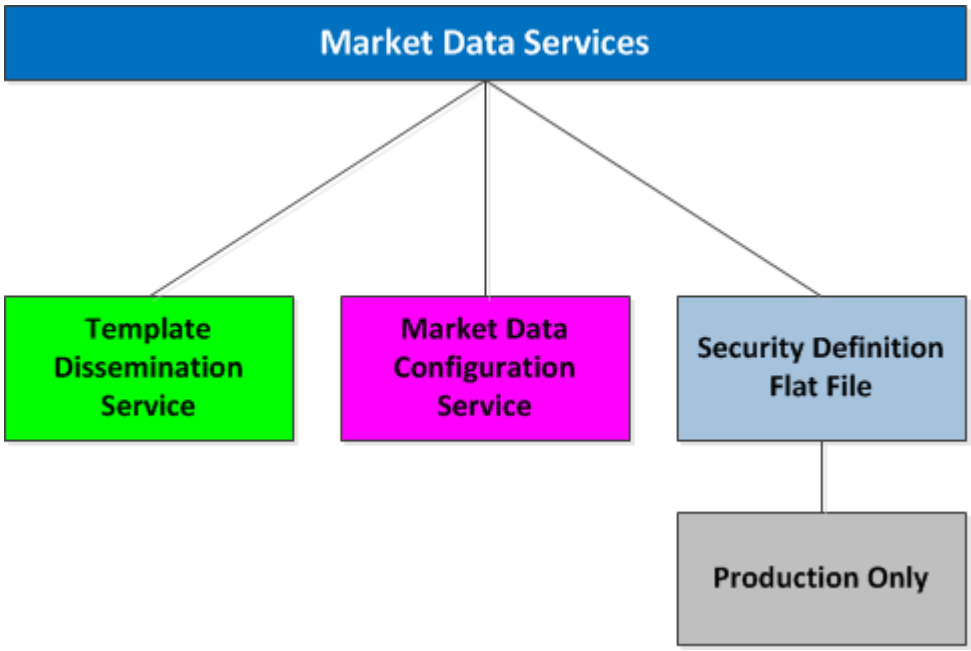
The TCP historical replay component allows you to request a replay of a set of messages already published on the UDP Incremental Market Data Channel or a snapshot message. The request specifies messages to be replayed. The request uses the Market Data - Request (tag 35-MsgType=V) message.

This type of request is sent through a new TCP connection established by the customer. The responses are sent by CME Group through this same connection and the connection is then closed by CME Group once the resend is complete. All responses are FIX/FAST encoded (including the reject response). Replay is limited to a maximum of 2000 messages, the recommended request level.

> This is not the preferred method for recovery (use Market Recovery).

Market Data Services

Market data services provide the external data required to process CME Group market data.



- Template Dissemination - method for client system to receive all active CME Group templates
- Market Data Configuration - allows client systems to receive list of all market data channel configurations
- Security Definition Flat File - contains all Security Definition (tag 35-MsgType=d) messages for CME Globex instruments

For additional information, Market Data - FTP Site Information.

Market Data Encoding - FAST Implementation

The FIX Adapted for STreaming (FAST) Protocol has been developed as part of the FIX Market Data Optimization Working Group. FAST is designed to optimize electronic exchange of financial data, particularly for high volume, low latency data dissemination. This topic describes implementation of FAST in receiving and processing CME Group FIX/FAST encoded electronic market data feed.

For more information see the FAST specification (version 1.x.1).

FAST is a data compression algorithm that significantly reduces bandwidth requirements and latency between sender and receiver. FAST works especially well at improving performance during periods of peak message rates. FAST extends the base FIX specification and assumes the use of FIX message formats and data structures. FAST is a standalone specification that uses templates to inform the receiver which operations to use in decoding. Templates allow FAST to achieve high levels of data compression with low processing overhead and latency compared to other compression utilities such as Zlib. The Preamble is found before the FAST encoded message, and contains the sequence number and subchannel identifier.

The following diagram shows the structure of a FAST encoded message:

Sequence #	Sub-Channel Identifier	PMAP	Template ID	Rest of Message
4 bytes	1 byte	*n* bytes	2 bytes	*n* bytes
Preamble		Encoded FAST message		

This topic describes concepts applicable to CME Group-specific FAST implementation; this topic is supplementary to the FAST specification referenced above.

See also:

- Presence Map and Stop Bit
- FAST Decoding

Preamble

The Preamble provides the sequence number and sub-channel identifier for the FAST encoded message. The Preamble consists of 5 non-FAST encoded bytes in Big Endian format that are found before all FIX/FAST messages on all feeds (Incremental, Market Recovery, Instrument Definition, and TCP Replay). The Preamble is found before the FAST encoded message, and contains the sequence number and sub-channel identifier. Processing of the Preamble is optional and FAST messages are not impacted by it.
Preamble Sequence Number

The Preamble sequence number is a 4 byte, unsigned integer representing the value distributed in tag 34-SeqNum of the related FAST message. The Preamble sequence number can be used during the arbitration process to determine whether any messages have been missed without decoding the FAST message, reducing processing time.
Preamble Sub-Channel Identifier

The sub-channel identifier is 1 unsigned byte found between the Preamble sequence number and the FAST message. It will be used to identify the sub-channel for all FAST messages. This makes it possible to process only the messages which carry market data for certain sub-channels and ignore all other sub-channels.

The sub-channel identifiers are mapped to group codes in the config.xml file.

The administrative message sub-channel will always be denoted as zero (0).

Stop Bit Encoding

Stop Bit encoding is a process incorporated in FAST that eliminates redundancy at the data field level by using a stop bit instead of the traditional separator byte. In FAST, a stop bit is used instead of the FIX traditional <SOH> separator byte. Thus 7 bits of each byte are used to transmit data and the eighth bit is used to indicate the end of a field.
FAST 7-Bit Binary Representation

FAST renders numbers into binary across the 7 data bits in each byte. Thus a number equal to or less than 2^7-1, (127) occupies one byte, a number between 2^7 and 2^7*2 – 1 (16,383), occupies two bytes, etc.

Implicit Tagging

In traditional FIX messages each field takes the form "Tag=Value<SOH>" where the tag is a number representing which field is being transmitted and the value is the actual data content. The ascii <SOH> character is used as a byte delimiter to terminate the field.

Example
35=x|268=3 (message header)
279=0|269=2|270=9462.50|271=5|48=800123|22=8 (trade)
279=0|269=0|270=9462.00|271=175|1023=1|48=800123|22=8|346=15 (new bid 1)
279=0|269=0|270=9461.50|271=133|1023=2|48=800123|22=8|346=12 (new bid 2)

FAST eliminates redundancy with a template that describes the message structure. This technique is known as 'implicit tagging' as the FIX tags become implicit in the data. A FAST template replaces the tag=value syntax with implicit tagging as follows:

- tag numbers are not present in the message but specified in the template
- fields in a message occur in the same sequence as tags in the template
- the template specifies an ordered set of fields with operators

Field Encoding Operators

FAST functions as a state machine and must know which field values to keep in memory. FAST compares the current value of a field to the prior value of that field and determines if the new value should be constant, default, copy, delta (integer or string), increment, or tail.
Dictionary Context

CME Group uses a dictionary context on a per-packet basis. A dictionary is a cache in which previous values are maintained. All dictionary entries are reset to the initial values specified after each UDP packet. Currently, CME Group sends one message per UDP packet.
Field Operators

> The following are general descriptions and may have exceptions depending upon the scenario.

A field within a FAST template may contain the Constant field operator. Please note that in some cases it is possible for a field to have no field operator.

- **Constant** – indicates that the field will always contain a predetermined value as specified by the value attribute. If the value is optional this field will contain a Pmap bit; if mandatory this field will not contain a Pmap bit.

> IMPORTANT NOTICE REGARDING FIELD OPERATORS
> The life cycle of the FIX/FAST protocol, including streamlined FIX/FAST, is rapidly approaching its end as an industry standard in the United States. Given the increased availability of bandwidth as well as high-speed connections such as co-location, compression has become less critical. In the short term, CME Group is removing certain field operators (copy, default, delta, and increment) from our data feed. In the longer term, we continue to evaluate a next-generation market data protocol.
>
> These operators have been removed from all production FIX/FAST channels on the Market Data Platform.
>
> New templates are required. The new templates are currently available in New Release and Production. CME Group recommends that customer systems pull the templates from the CME Group ftp site every week, prior to Sunday startup.

FAST Data Types

A field within a FAST template will have one of the following Data Types indicating the required decoding action:

- **String** – used to represent ASCII or Unicode values using the FAST 7-bit binary encoding.
- **Signed Integer** – used to represent a signed (+/-) integer using the FAST 7-bit binary encoding. A twos complement integer representation is used, with the most significant data bit being the sign bit. Note that contiguous leading bits of the same value must be dropped (so, for instance, only 1 byte is required to encode -1, 0xFF). In some cases, a 0x00 byte will be the most significant byte sent to preserve sign, so 64 is represented 0x00 0xC0.
- **Unsigned Integer** – used to represent unsigned integers using the FAST 7-bit binary encoding.

- **Decimal** – used to represent a floating point number as exponent and mantissa. The exponent is a signed integer used to express precision and the mantissa is a signed integer used to express the value. The numerical value is obtained by multiplying the mantissa with the base-10 power of the exponent expressed as:

number = mantissa * 10exp

The exponent and mantissa can be encoded as a single, composite field or as individual conjoined fields.

FAST Template

A FAST template corresponds to a FIX message type and uniquely identifies an ordered collection of fields. The template also includes syntax indicating the type of field and transfer decoding to apply. A template is communicated between CME Group and client systems in XML syntax using the FAST v1.1 Template Definition Schema maintained by FIX. The XML format is human- and machine-readable and can be used for authoring and storing FAST templates. Session Control Protocol (SCP) will not be used.

Presence Map and Stop Bit

- Presence Map Rules
- Message Structure
- Template ID Usage
- Optional vs Mandatory Fields
- XML Template Example

The Presence Map (Pmap) indicates which fields in the template have data present and which fields have data implied. A Pmap is a sequence of encoded bits with each bit representing a template field according to sequence. Fields with data present have the Pmap bit set to '1'. Fields with data implied have the Pmap bit set to '0' (Exception: fields with a Constant operator in which the bit is set to '1' for an implied state). A Pmap occurs at the beginning of each FAST message and at the beginning of any sequence/group of fields as long as those fields are defined such that a Pmap slot is required.

Presence Map Rules

Pmap rules determine when a FAST template field requires a corresponding Pmap bit. A field does not require a bit in the pmap when it meets any of the following criteria:

1. The field is defined as mandatory without a field operator – a value will always be present.
2. The field is defined as mandatory with a constant operator – a value should always be instantiated in the decoded message based on the value in the template.
3. The field is defined as mandatory with a delta operator – a delta value is always present.
4. The field is defined as optional without a field operator – either a value or NULL will always be present.
5. The field is defined as optional with a delta field operator – a delta value or NULL will always be present.

See the FAST specification (version 1.x.1) for complete information on presence map rules.

Rules for Determining if a Presence Map Bit Is Required

Operation	Mandatory	Optional
none	no	no
constant	no	yes
copy	yes	yes

default	yes	yes
delta	no	no
increment	yes	yes

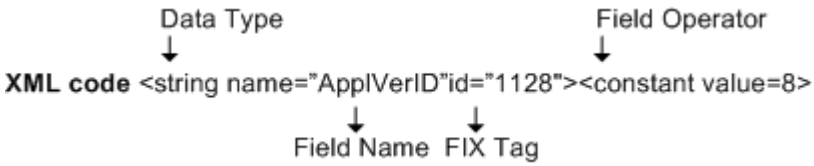
Message Structure

> The Market Data Incremental Refresh (tag 35-MsgType = X) message is used for example purposes; not all FAST messages follow this format.

The FIX Market Data Incremental Refresh (tag 35-MsgType = X) message is made up of three components: a Header, a Body and a set of one or more Market Data Entries as shown in the diagram below. The Header carries transmission details. The Body carries information pertinent to all entries in the message such as TradeDate. The Market Data Entry carries specific instructions for updating the book or recording trades. The templates provided by CME Group conform to this general structure and use only fields that are part of the Market Data Incremental Refresh (tag 35-MsgType = X) message.

Header tag 35-MsgType|tag 1128-ApplVerID|tag 34-MsgSeqNum|tag 52-SendingTime
Body tag 75-TradeDate|tag 268-NoMDEntries
Market Data Entry tag 279-MDUpdateAction|tag 269-MDEntryType|Tag 48-SecurityID|tag 22-SecurityIDSource|tag 270-MDEntryPx|tag 346-NumberOfOrders

Example: How to Read a Template Field
Field Operators and Data Types appear within the Template as follows:



Template ID Usage

Each template is assigned a Template ID to uniquely describe the format of an encoded message. A Template ID is carried in every encoded message to provide a reference to the correct template for decoding purposes. The Template ID is an unsigned integer carried as the first data field following the first Pmap of every message allowing the decoding system to apply the correct template to the message upon receiving it.

CME Group will not send the same Template ID more than once in a UDP packet. When the packet is received, the sequence to begin decoding a message is:

1. Locate and read the first Pmap field, which begins in the first first byte of the application

data preceding the MDP/FIX Header.
2. Treat the next field as the Template ID (the first logical field of the message).
3. Retrieve the template specified by the Template ID.
4. Decode the remainder of the message.

Optional vs Mandatory Fields

A field is defined as optional when it is possible for a value to not be present in the decoded message. When a field is optional, the wire representation "null" indicates the field is not present. A field is defined as mandatory when that field must be present in the decoded message. Note that a field may not be present ("coded away") in the wire-formatted message but can be present in the decoded message as the result of an encoding operation (e.g., default) that removes data for efficiency but continues to imply data is present through the Pmap.

Optional fields are useful when a generic template is used to provide multiple market data types such as book updates and trades. In this situation, there may be fields which are present in one occurrence of the repeating group but not in another within a given message.

For example, the MDEntries repeating group that is present in the MDIncRefresh template shown above can be used to express a trade, bid, ask, high, low, etc. within a single message. A trade entry will not use tag 346, NumberOfOrders, which will then be defined as optional. However, FAST requires that this field be accounted for in the encoded message if specified in the template. This is done through the use of a reserved value of NULL to indicate that the field is not present in the decoded data. In the serialization layer, FAST reserves binary zeros to indicate a NULL value which tells the decoder that no data is present for this template distribution.

> CME Group requires that client systems use an API layer to load templates rather than implement hard coded templates. Since templates change, this will facilitate template modification in production environments.

Rules for Determining if a Field Is Nullable

Operation	Mandatory	Optional
none	no	yes
<Constant/>	no	no
<Copy/>	no	yes
<Default/>	no	yes
<Delta/>	no	yes
<Increment/>	no	yes

XML Template Example

A template consists of Field Instructions that define the fields contained in the message. Field Instructions specify the field name, tag number, data type, field operator, and presence attribute that indicates if a field is optional or mandatory.

A sample market data template is shown below. The syntax is standard XML and can be parsed using a variety of open source tools. Valid template syntax is determined by the FAST Template Schema which is available in the FAST specification (version 1.x.1).

- The real-time feed templates are based on the Market Data Incremental Refresh message type.

Client FAST Decoders read the template file to recognize how each field is encoded in terms of data type representation (Transfer) and data redundancy removal (Field).

The template is constructed of several sections including Template Identification, Header, Body and Sequence. Template Identification provides the template name and identifier. The Header includes FIX header fields such as ApplVerID (tag 1128), MsgType (tag 35), and SendingTime (tag 52). The Body provides information common across all repeating groups. Sequence represents a repeating group with a corresponding length field and a set of repeating group fields which carry the detailed entry information.

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <template name="MDIncRefresh" id="30" xmlns="http://www.cme.com/mdp/fast-templates" >
3    <typeRef name="MDIncRefresh"/>
4    <string name="ApplVerID" id="1128"> <constant value="7"/> </string>
5    <string name="MessageType" id="35"> <constant value="X"/> </string>
6    <string name="SenderCompID" id="49"> <constant value="CME"/> </string>
7    <uInt32 name="MsgSeqNum" id="34"> </uInt32>
8    <uInt64 name="SendingTime" id="52"> <delta/> </uInt64>
9    <uInt32 name="TradeDate" id="75"><copy/> </uInt32>
10   <sequence name="MDEntries"><length name="NoMDEntries" id="268"></length>
11     <decimal name="MDEntryPx" id="270">
12       <exponent><copy value="-2"/></exponent>
13       <mantissa><delta/></mantissa> </decimal>
14     <int32 name="MDEntrySize" id="271"> <delta/> </int32>
15     <uInt32 name="MDEntryTime" id="273"> <delta/> </uInt32>
16     <uInt32 name="MDPriceLevel" id="1023" presence="optional"> <increment/> </uInt32>
17     <uInt32 name="MDUpdateAction" id="279"> <copy value="1"/> </uInt32>
18     <string name="MDEntryType" id="269"> <copy value="0"/> </string>
19     <uInt32 name="SecurityID" id="48"> <copy/> </uInt32>
20     <uInt32 name="SecurityIDSource" id="22"> <constant value="8"/> </uInt32>
21     <int32 name="NumberOfOrders" id="346" presence="optional"> <delta/> </int32>
22     <string name="QuoteCondition" id="276" presence="optional"> <copy value="K"/> </string>
23     <string name="TickDirection" id="274" presence="optional"> <default/> </string>
24     <int32 name="NetChgPrevDay" id="451" presence="optional"> <default/> </int32>
25     <string name="TradeCondition" id="277" presence="optional"> <default/> </string>
26     <int32 name="TradeVolume" id="1020" presence="optional"> <default/> </int32>
27     <string name="TradingSessID" id="336"><default value="1"/> </string>
28   </sequence>
29 </template>
30
```

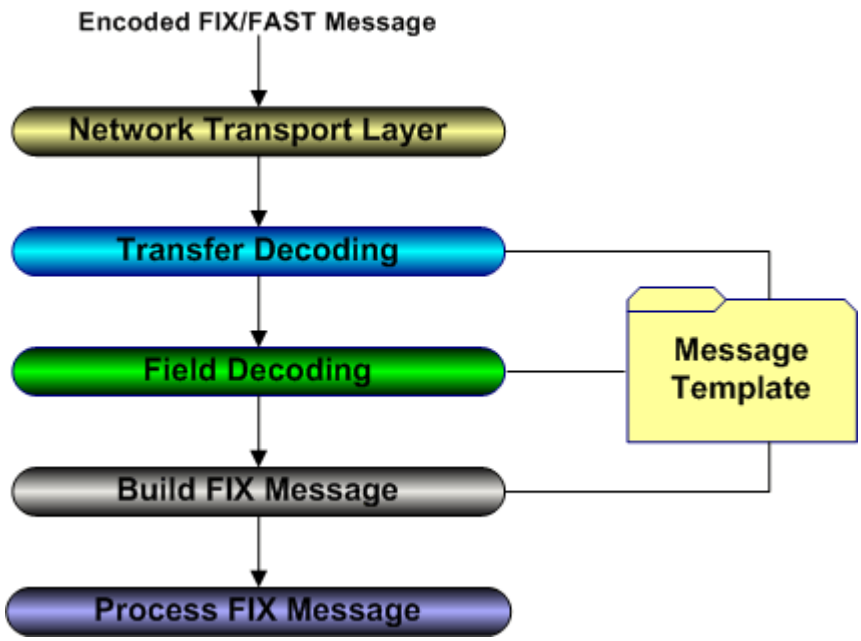
FAST Decoding

This section presents one possible approach to decoding CME Group FAST messages to demonstrate the fundamental concepts involved in the process. This approach is provided for example purposes and is not definitive.

FAST processing decodes a FIX/FAST message by means of FAST templates provided by CME Group. The FAST template contains the instructions to decode and reconstruct compressed message data into the FIX format and also supports repeating groups (sequences) that allow a single message to convey multiple instructions (i.e. book update, trade, high/low, etc.).

> CME Group FAST encoding is bit-specific to the FAST message specification. CME Group FAST encoding always uses the global dictionary.

A decoding sequence can follow the steps diagrammed below; this topic describes Transfer Decoding, Field Decoding, and Building the FIX message only.



1. Transport - Client system receives encoded FAST message.
2. Transfer Decoding
    - Identify template
    - Extract binary encoded bits
    - Map bits to fields per template
3. Field Decoding - Apply operators to determine values per template field.
4. Build FIX message.
5. Process FIX message.

Market Data - Decoding Sequence

This section provides a detailed example of the process for decoding a FAST message. You should thoroughly test your application to determine that you have properly decoded a message.

> This decoding example is a generic description of the basic decoding process; it does not reflect the structure of the reference code.

FAST message decoding consists of three major steps:

Step 1. Transfer Decoding

- Identify Template ID and Pmap bit value for each given field. Pmap values are 0, 1.
- Decode FAST 7-bit binary values to identify data present in the message.

Step 2. Field Decoding

- Apply field operators to extracted binary values.
- Determine state of field to be decoded.

Step 3. Build FIX Message

- Apply FIX or internal structure to decoded message

The decoding sequence in the example used throughout this section uses a FAST message containing a header, trade, and two new bids.

See also:

- Market Data - Transfer Decoding
- Market Data - Decoding Process
- Market Data - Build the FIX Message

Market Data - Transfer Decoding

Upon receipt of the FAST message, the client decoder loads the template according to the Template ID. From this template the client decoder identifies how to decode each field, whether a field will have a Pmap bit, and if a field can carry a null value. In this example the MDIncRefresh template contains a header and repeating group structure (italics).

The following template is an example template:

```
<template name="MDIncRefresh" id="30">
  <typeRef name="MDIncRefresh"/>
<string name="MessageType" id="35"> <constant value="X"/> </string>
<sequence name="MDEntries"><length name="NoMDEntries" id="268"></length>
  <uInt32 name="MDUpdateAction" id="279"> <copy value="0"/> </uInt32>
  <uInt32 name="MDEntryType" id="269"> <copy value="0"/> </uInt32>
  <decimal name="MDEntryPx" id="270">
    <exponent><copy value="-2"/></exponent>
    <mantissa><delta/></mantissa> </decimal>
<int32 name="MDEntrySize" id="271"> <delta/> </int32>
<uInt32 name="MDPriceLevel" id="1023" presence="optional"> <increment/></uInt32>
<uInt name="SecurityIDSource" ID="22"><constant value="8"/></uInt32>
<int32 name="NumberOfOrders" id="346" presence="optional"> <delta/> </int32>
  </sequence>
</template>
```

- Field Decoding Overview
- When to Reset Decoder State
- Receiving Data over MDP
- Decoding a Packet
- Decoding Messages in a Packet
- Error Handling in a Broadcast Environment

Transfer Decoding is the initial step that converts data from the FAST 7-bit binary format.

> The decoding process can take place on a field-by-field basis, de-serializing and rebuilding the decoded value for each field. Alternately, the decoding process may de-serialize the entire message and then make a second iteration over the message to rebuild the decoded values.

The FAST transfer format is a binary representation enhanced with the following attributes:

- **Stop Bit** - Each data byte contains a stop bit to indicate whether this is the last byte of a field. The stop bit is the seventh bit of the most significant byte. A stop bit set to 1 indicates the last byte of a field. A stop bit

set to zero indicates this is not last byte of the field.
- **Presence Map** - The Presence Map occurs at the start of every message or repeating group and indicates the presence or absence of individual fields. The template specifies the type of transfer decoding to employ for each field in the message (e.g. string, integer, decimal, etc.)

> ⚠ Warning
> FAST does not support timestamps. CME Group will convert the timestamp to an integer by removing punctuation from the UTC timestamp prior to encoding. The decoding application should convert the integer to the FIX UTC format after decoding. In addition, the timestamp header on the message packet should not be used for latency comparisons. The message must be decoded to determine the timestamp for updates.

Field Decoding Overview

Field decoding is the second part of the decompression process that reconstructs data values according to template-specified operations. Field decoding operations are assigned per field within the template; decoding reinstates data as indicated by the template.

When to Reset Decoder State

The state of the decoder must be reset for each received UDP packet and the dictionary applied. Resetting state means that all fields are set to their initial pre-processing state. This is because UDP transport is not reliable and data in a packet cannot be dependent on data in a previous packet since it is possible for packets to be lost or arrive out of sequence.

Receiving Data over MDP

Currently, CME Group will send only a single FIX message per packet; however, in future implementations CME Group may send multiple messages per packet. The basic operation requires the decoding of data within a discreet packet as well as the ability to determine the end of one message and beginning of another within that packet.

Decoding a Packet

Decoding of the datagram should be conducted as a unit of work in which only the data in a given packet participates in the scope of the decoding process. To determine when the process has fully decoded a discreet message, it is important to use the template to step through a message field-by-field.

Decoding Messages in a Packet

When the decoder determines that it has processed all repeating groups in the message, and that there are no further fields specified by the template, then it can conclude that the end of the message has been reached. In this way, it is necessary to use the presence maps, NoMDEntries, and the template to determine when end of message has been reached. To recap, the steps involved are:

- Decode fields in the non-repeating body of the message using the high-level presence map.
- If the last field in the template has been reached then end of message has been encountered.
- If the NoMDEntries field is present, then process each repeating group using the repeating group presence
  map and template.
- When number of repeating groups processed equals the count specified in NoMDEntries, then the end of the repeating groups has been encountered.

Error Handling in a Broadcast Environment

If FAST encounters an error during the process of decoding the contents of a UDP packet, the process should stop, throw an exception, and decode the next packet. In all likelihood, the error is due to an inconsistency between the encoded data and the template being used to decode the data. At this point, it is advised that parties synchronize templates in order to ensure that the same data definitions are used.

Market Data - Decoding Process

The client system reads the Pmap to identify required field-level decoding.

Pmap 1 11000000	Header Fields	Pmap 2 10101100	Trade Fields	Pmap 3 10101000	New Bid 1 Fields	Pmap 4 10000000	New Bid 2 Fields

In this example light grey values indicate an unused bit.

Step 1. Extract Pmap1 from data. Pmap1 is the top-level Pmap that contains the Pmap bits for all fields requiring a Pmap bit in the message.

- The first bit in Pmap1 after the stop bit (in bold) is assigned to the Template ID, which always has a bit. The bit is turned on indicating that the Template ID is present in the encoded message and that the corresponding template must be loaded.
- The first field in the template is the MessageType, defined as a mandatory field using a Constant operator with a value of "X" retrieved from the template and therefore does not have a bit.
- The next field in the template is NoMDEntries. The length specified in the sequence determines the number of repeating groups in the message. This field does not have a bit since the field is mandatory and does not have an operator.
- **The remaining bits in Pmap1 are defaulted to 0 but are not referenced by the decoding process; at this point, the Header fields have been decoded.**

Step 2. Extract Pmap2 from data. Pmap2 represents the first repeating group and contains trade-related fields.

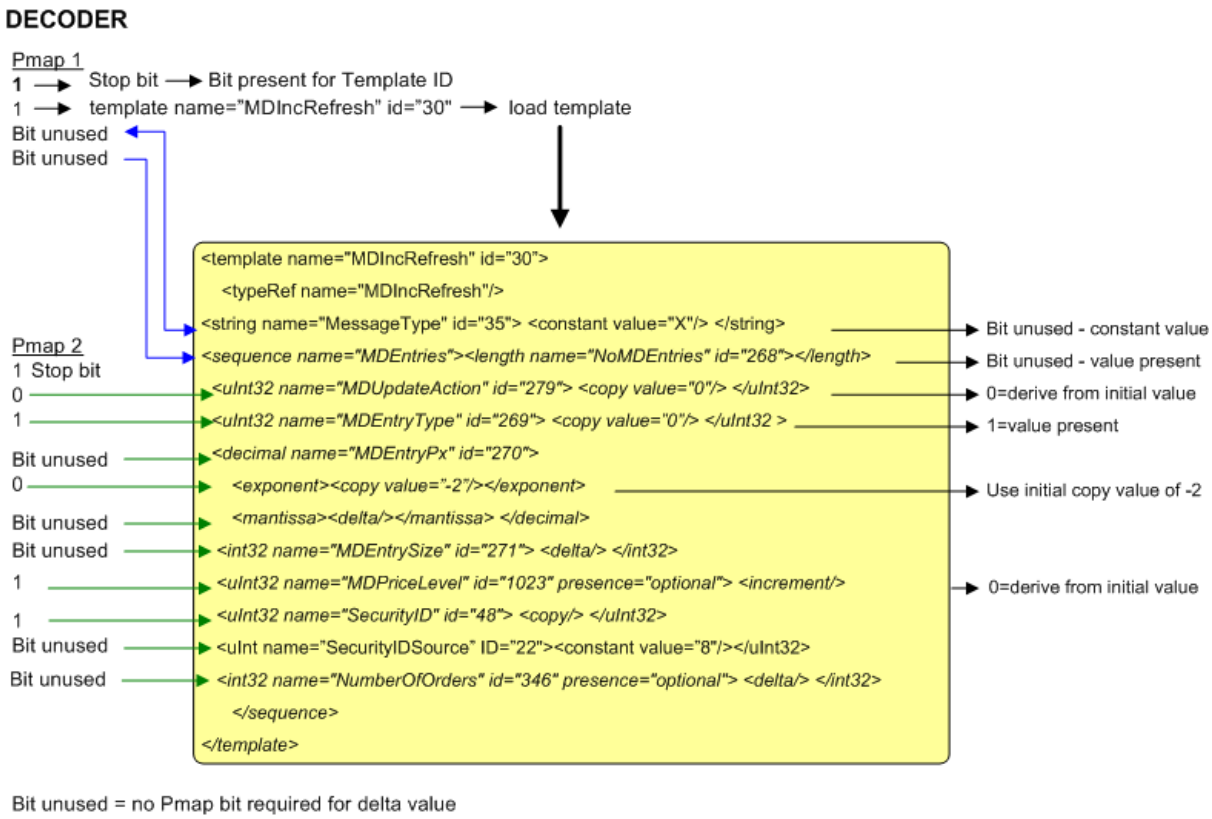
- The next field in the template is MDUpdateAction. By definition, this field requires a bit in the Pmap since it uses the Copy field operator. In this example, the bit is set to 0 indicating that no data is present and that the value must be derived from the initial value.
- The next field in the template is MDEntryType, which is a Copy field. The Pmap set to '1' indicates an encoded value present in the field.
- The decoding process continues using the template and Pmap2 to traverse the data.

Step 3. Extract Pmap3 from data. Pmap3 represents the presence map for repeating group2 followed by the New Bid1 fields.

- By the time the decoding process reaches Pmap3, prior values have been established for several fields based on the content of the prior repeating group.
- The decoding process uses the prior values in combination with the encoded data to generate the proper decoded values.

Step 4. Extract Pmap4 from data. Pmap4 is the final presence map and the decoder recognizes that all bits are set to off indicating that either prior values should be used or that fields do not use a bit.

The following diagram illustrates the decoding process flow:



Market Data - Build the FIX Message

The table in this example shows the step-by-step conversion for each field of the FAST message into the
FIX format. The following steps describe the actions taken to obtain the data shown in the Message Decoding Process
table.

1. Decoding begins with the identification of the Pmap bit for each field.
2. The encoded FAST 7-bit binary values are obtained as shown in the "Encoded FAST 7-Bit Binary Value" column.
3. The encoded FAST 7-bit binary values from step 2 are deserialized based on the data type specified in the template.
4. The decoder maintains the state of prior values for each field throughout decoding and applies them for fields having operators of Delta, Copy, or Increment.
5. Obtain fully decoded values.

Edit Document

Message Decoding Process Table									
				Step 1	**Step 2**	**Step 3**	**Step 4**		**Ste**
#	**Field Name**	**Presence***	**Data Type/Field Operator**	**Pmap Bit**	**Encoded FAST Hex/Binary Value**	**Deserialized Encoded Value**	**Prior Value**	**Initial Value**	**Dec Val**
BIT STREAM = 11000000 10011110 10000011									
1	Template ID	M	uint32/copy	1	0x9e – 10011110	30			TID

2	MsgType	M	string/constant	No bit	None	None		X	35=
3	NoMDEnt	M	uint32/no operator	No bit	0x83 - 10000011	3			268
Repeating Group 1 - Trade									
BIT STREAM = 10101100 10000010 00111001 01100000 11001010 10000101 10000000 00110000 01101010 11111011 1000(									
4	UpdateAction	M	uint32/copy	0	None	None		0	279
5	EntryType	M	uint/copy	1	0x82 - 10000010	2		0	269
6	EntryPrice	M	Exp: int32/copy Mant: int64/delta	Exp - 0 Mant – No bit	Exp: None Mant: 0x39 0x60 0xca – 00111001 01100000 11001010	Exp = None Mantissa = 946250		-2	270
7	EntrySize	M	int32/delta	No bit	0x85 - 10000101	5			271
8	PriceLevel	O	uint32/increment	1	0x80 – 10000000	NULL		1	102 Valu
9	SecurityID	M	uint32/copy	1	0x30 0x6a 0xfb - 00110000 01101010 11111011	800123			48=
10	SecurityIDSource	M	uint32/constant	No bit	None	None		8	22=
11	NoOrders	O	int32/delta	No bit	0x80 – 10000000	NULL		0	346 Valu
Repeating Group 2 – New Bid 1									
BIT STREAM = 10101000 10000000 11001110 00000001 10101010 10000010 10010000									
12	UpdateAction	M	uint32/copy	0	None	None	0		279
13	EntryType	M	uint/copy	1	0x80 - 10000000	0	2		269
14	EntryPrice	M	Exp: int32/copy Mant: int64/delta	0 No bit	Exp: None Mant: 0xce - **1** 1001110	Exp = None Mantissa = -50	-2 946250		270
15	EntrySize	M	int32/delta	No bit	0x01 0xaa – **0** 0000001 **1** 0101010	170	5		271
16	PriceLevel	O	uint32/increment	1	0x82 - **1** 0000010	2	NULL		102
17	SecurityID	M	uint32/copy	0	None	None	800123		48=
18	SecurityIDSource	M	uint32/constant	No bit	None	None	8		22=
19	NoOrders	O	int32/delta	No bit	0x90 - 10010000	16*	0**		346
Repeating Group 3 – New Bid 2									
BIT STREAM = 10000000 11001110 11010110 11111101									

20	UpdateAction	M	uint32/copy	0	None	None	0		279
21	EntryType	M	uint/copy	0	None	None	0		269
22	EntryPrice	M	Exp: int32/copy Mant: int64/delta	0 No bit	0 No bit	Exp: None Mant: 0xce - 11001110	Exp = None Mantissa = -50	-2 946200	270
23	EntrySize	M	int32/delta	No bit	0xd6 – 11010110	-42	175		271
24	PriceLevel	O	uint32/increment	0	None	None	1		102
25	SecurityID	M	uint32/copy	0	None	None	800123		48=
26	SecurityIDSource	M	uint32/constant	No bit	None	None	8		22=
27	NoOrders	O	int32/delta	No bit	0xfd - 11111101	-3	15		346

*M=mandatory O=optional
**Subtract 1 from non-negative optional integer fields
***For delta fields, if a null is received, the prior value in the dictionary is not changed. In this case, the intital value was not specified in the template, so the initial value is set to '0'. The first encoded field received has a value of null, so the prior value is '0' when decoding the second instance of this field.

Result - Decoded Values

35=x|268=3 (message header)
279=0|269=2|270=9462.50|271=5|48=800123|22=8 (trade)
279=0|269=0|270=9462.00|271=175|1023=1|48=800123|22=8|346=15 (new bid 1)
279=0|269=0|270=9461.50|271=133|1023=2|48=800123|22=8|346=12 (new bid 2)

Market Data - System Startup

This section contains an overview of the Sunday startup process for the production environment.

Pre-Opening Startup

For a startup prior to the weekly market open, all market data including book updates, statistics, quotes, instrument definitions, trades, instrument/instrument group, status) will be disseminated through the Incremental UDP Feed A and Feed B. Book updates include bid/ask/trade. Statistics include high trade, low trade, volume, and settlement. Instrument definitions include the list of all tradable instruments. Follow the process below to ensure that all necessary market data is received:

1. Download the configuration files and template files from the ftp site. Refer to Market Data - FTP Site Information for more information.
2. Listen to the Incremental feed for incremental market data and start normal processing.

The start of a pre-open market state is defined by receipt of the Security Status (tag 35-Msg-Type=f) message where tag 326-SecurityTradingStatus=21 (Pre-Open). Any market

participant who initiates the entry of an order for a Trades At Settlement (TAS) or Trade At Market (TAM) product, prior to receipt of the pre-open market status message, will be subject to disciplinary action by a panel of the Business Conduct Committee, notwithstanding that the order may have been rejected by the CME Globex system.

Late Joiner Startup

For a late joiner startup, follow the process below to ensure that all necessary market data is received:

1. Download the configuration files and template files from the ftp site. Refer to Market Data - FTP Site Information for more information.
2. Listen to the Instrument Definition feed.

   Refer to Market Data - Instrument Replay Overview for more information.
3. Listen to the Incremental feed for incremental market data. Begin the natural refresh process and begin queuing messages.

   > The incremental market data may complete a natural refresh (liquid instruments only) that would construct the current, correct state of a book. Refer to Market Data - Natural Refresh for more information.
4. Listen to the Market Recovery feed for the latest snapshots.

   Use the latest snapshot to verify that the book was correctly created via natural refresh and to retrieve the latest statistics. When the latest snapshots are received, the value for tag 369-LastMsgSeqNumProcessed in the snapshot matches tag 34-MsgSeqNum in the incremental feed message. Also, the value for tag 83-RptSeq in the snapshot matches tag 83-RptSeq in the incremental feed.

   If the book for an instrument was not completely constructed using natural refresh, then apply the snapshot. Start discarding queued messages from the incremental feed until tag 34-MsgSeqNum in the message has the same value as tag 369-LastMsgSeqNumProcessed in the snapshot. The discarded messages contain information that was already included in the snapshot message.

   > Information for instruments included for the first time in the latest incremental feed message (with 34-MsgSeqNum equal to tag 369-LastMsgSeqNumProcessed in the snapshot) may not be in the latest snapshot. Apply the latest incremental feed message to obtain this information.
5. Stop listening to the Market Recovery and Instrument Definition feeds.
6. Start normal processing.

Heartbeats

When the Market Recovery and Instrument Definition components begin, CME Globex sends Heartbeat (tag 35-MsgType=0) messages at a predefined interval on the Market Recovery and Instrument Definition feed. The components begin heartbeating on Sunday from the time they start until the time the first data is sent. When the Market Recovery and Instrument Definition channels start looping, CME Globex no longer sends Heartbeat tag 35-MsgTpe=0 messages.

Market Data - Incremental Feed Arbitration

> CME Group strongly recommends that client systems process both the Incremental Feed A and Incremental Feed B due to the unreliable nature of UDP transport.

Dual transmission on Incremental Feed A and Incremental Feed B to allows client systems to arbitrate between feeds to ensure message sequence integrity.

Any given message sent on both feeds can arrive first on either feed depending upon network carrier and independent of CME Group.

The following represents the CME Group best practice for client system arbitration across Incremental Feed A and Incremental Feed B.

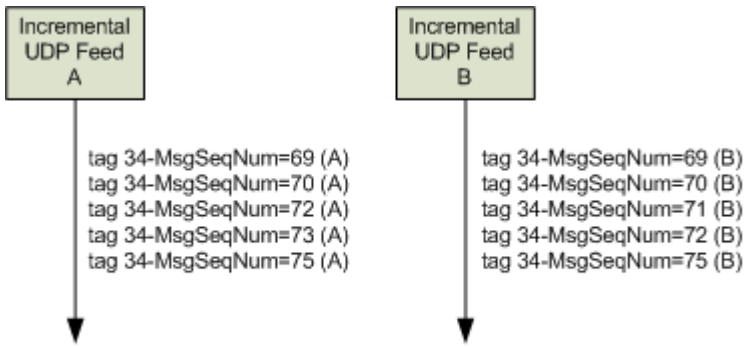
> The Preamble sequence number, or tag 34-MsgSeqNum, can be used during the arbitration process to determine whether any messages have been missed without decoding the FAST message. Using the Preamble sequence number requires less processing than using tag 34-MsgSeqNum.

Arbitration Process Overview

Client systems should use the A and B Incremental Feeds to arbitrate as follows:

1. Listen to messages from both Incremental Feed A and Incremental Feed B.
2. Process messages by incremental sequence number (tag 34-MsgSeqNum).
3. Discard a message if the sequence number (tag 34-MsgSeqNum) has already been processed.
4. If a sequence number (tag 34-MsgSeqNum) gap is detected, this indicates a packet was lost on both the Incremental Feed A and Incremental Feed B. In this scenario client systems must initiate the recovery procedure.

Sequence Gap Example



Message Processing Example

Messages are received at client system from Incremental Feed A and Incremental Feed B individually.

1. Receive message on feed A - tag 34-MsgSeqNum=69.
   - Process message on feed A - tag 34-MsgSeqNum=69.
2. Receive message on feed B - tag 34-MsgSeqNum=69.
   - Discard message on feed B - tag 34-MsgSeqNum=69.
3. Receive message on feed A - tag 34-MsgSeqNum=70.
   - Process message on feed A - tag 34-MsgSeqNum=70.

4. Receive message on feed B - tag 34-MsgSeqNum=70.
    - Discard message on feed B - tag 34-MsgSeqNum=70.
5. Receive message on feed B - tag 34-MsgSeqNum=71.
    - Process message on feed B - tag 34-MsgSeqNum=71.
6. Receive message on feed B - tag 34-MsgSeqNum=72.
    - Process message on feed B - tag 34-MsgSeqNum=72.
7. Receive message on feed A - tag 34-MsgSeqNum=72.
    - Discard message on feed A - tag 34-MsgSeqNum=72.
8. Receive message on feed A - tag 34-MsgSeqNum=73.
    - Process message on feed A - tag 34-MsgSeqNum=73.
9. Receive message on feed A - tag 34-MsgSeqNum=75.
    - Discard message on feed A - tag 34-MsgSeqNum=75.
10. Receive message on feed B - tag 34-MsgSeqNum=75.
    - Discard message on feed B - tag 34-MsgSeqNum=75.
    - Gap detected because last message processed was tag 34-MsgSeqNum=73 on both feed A and feed B.
    - Begin recovery process - tag 34-MsgSeqNum=75.

Market Data - Services

There are three services available: Template Dissemination Service, Market Data Configuration Service, and Security Definition Flat File.

Service	Description
Template Dissemination	FIX/FAST is a template-based protocol by which a given message is interpreted by means of its corresponding template. Each message contains a unique Template ID that references the template to use to interpret the message. The template dissemination service provides a method for client systems to receive all of the CME active templates, or the templates associated with either a Template ID or a Market Data Group.
Market Data Configuration	The Market Data Configuration Service allows clients systems to receive the list of all Market Data Channel configurations (multicast IP, product group, Security ID, and sub-channel identifier). ⓘ Product Groups can be configured for more than one channel.

Security Definition Flat File	The Security Definition Flat File (secdef.dat) is available only for the Production environment. The process to generate the file begins each Sunday through Friday as follows: All start times are Central Time (CT) • Monday - 8:55 AM • Sunday - 12:15 PM • Sunday through Friday - 2:30 PM • Monday through Friday - 8:30 PM ⓘ The availability of the file will vary. Typically it takes approximately 50 - 60 minutes to complete the creation of the file. The Security Definition Flat File file contains the Security Definition (tag 35-MsgType=d) message information for the start of the next trading day, including instrument data and daily settlement prices. This file is not refreshed throughout the trading day. Once a trading day has started, the following intra-day instrument changes are not reflected in the file until a new one is generated that evening: • New instruments added during the day • Covered options strategies that expire at the end of the trading day • User-Defined Spreads (UDS) created during the trading day

ⓘ CME Group strongly recommends using this file either in addition to or as a back up for the market data channels due to risk of unaccounted for intraday instrument additions and deletions.

Market Data - FTP Site Information

An FTP site (ftp://ftp.cmegroup.com) is used to disseminate templates and market data configuration information. This FTP site contains the template files and configuration files for all environments and the Security Definition Flat File for the Production environment. Template and market data configuration details for the production environment are only available to customers after the certification process is complete.
Information applies as follows in the table:

- Environment - specific environment (i.e., Certification, New Release, Production).

ⓘ The AutoCert+ tool will indicate which environment you need to connect to for certification. For additional information on AutoCert+ access, refer to the AutoCert+ Access Guide.

- Service - the Template or Configuration service.
- FTP Site - address of FTP site.
- Directory Location - identifies the directory.
- Client System Update Schedule - Client systems should download updates according to the schedule specified.

Environment*	Service	FTP Site	Directory Location	Client System Update Schedule

Certification	Template	ftp.cmegroup.com	/Cert/Templates	Sunday prior to market open
Certification	Configuration		/Cert/Configuration	daily
Certification AutoCert+	Template		/CertAutoCertPlus/Templates	Sunday prior to market open
Certification AutoCert+	Configuration		/CertAutoCertPlus/Configuration	daily
New Release Certification	Template		/NRCert/Templates	Sunday prior to market open
New Release Certification	Configuration		/NRCert/Configuration	daily
New Release Certification AutoCert+	Template		/NRAutoCertPlus/Templates	Sunday prior to market open
New Release Certification AutoCert+	Configuration		/NRAutoCertPlus/Configuration	daily
Production	Template		/Production/Templates	Sunday prior to market open
Production	Configuration		/Production/Configuration	daily
Production	Flat File		/Production	8:30pm Sunday -Thursday

There is also an ftp site on the CME Group network that can be accessed as an alternative to the public ftp.cmegroup.com site. Please contact your Global Command Center (GCC) for additional information.

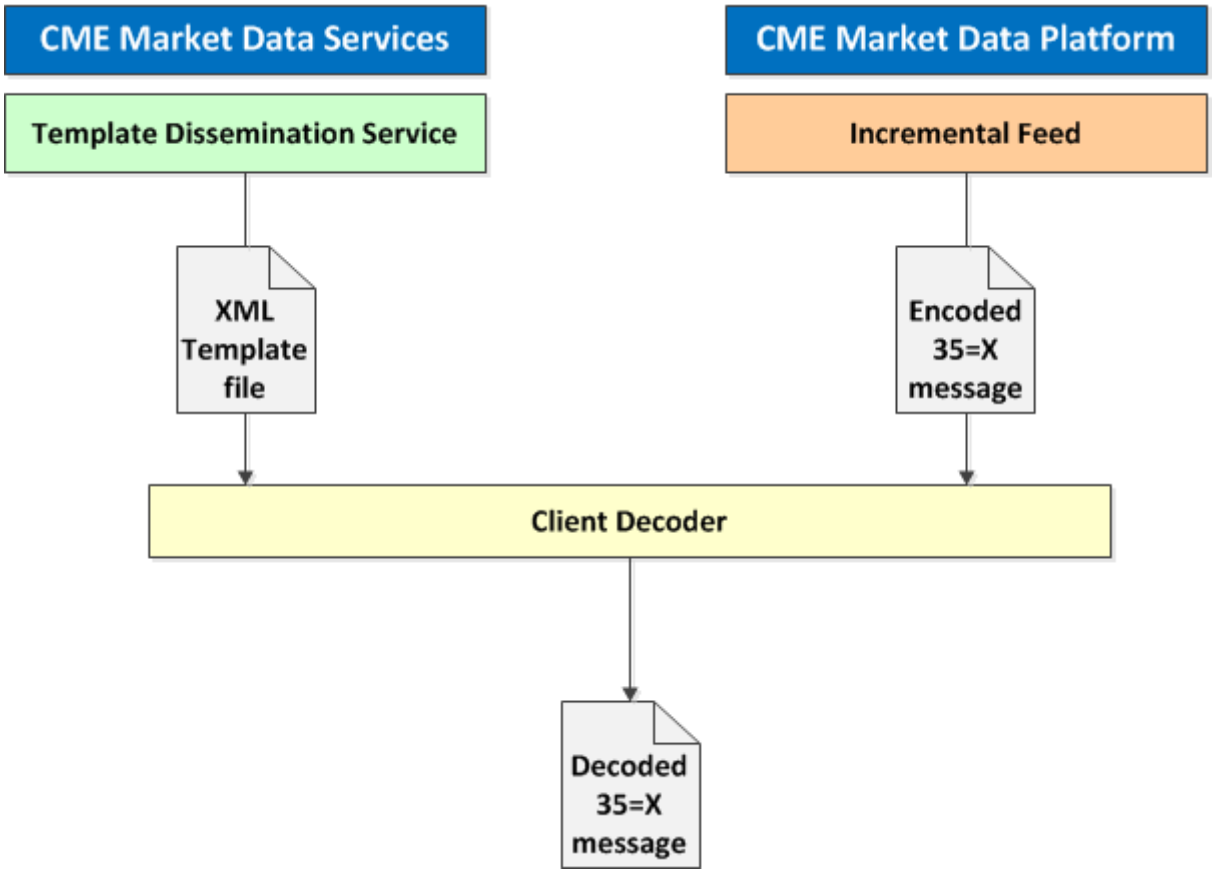
For MexDer markets, in the New Release and Production environments, the configuration file is updated daily, beginning Monday morning prior to market open.

Market Data - Template Overview

CME Group provides a single xml file that contains a collection of templates and is shared across all market data channels between the encoder (CME Group side) and decoder (customer side).

Each template has a unique template ID that describes the format of an encoded message. A template ID is carried in every encoded message to provide a reference to the correct template for decoding purposes.

The template ID is an unsigned integer carried as the first data field following the first Pmap of every message allowing the decoding system to apply the correct template to the message upon receiving it.



A template selection process should be carried out on the customer side. The decoder should dynamically identify and retrieve the correct template indicated in the encoded message, then use that template to decode the message.

Example: Template ID

```
2  <template name="MDIncRefresh" id="35" xmlns="http://www.fixprotocol.org/ns/fast/td/1.1">
3    <typeRef name="MDIncRefresh"/>
```

CME Group will not send the same template ID more than once in a UDP packet.

Template Distribution

The current template for each environment is available for download from the CME Group ftp site.

Refer to Market Data - Services for an outline of the process.

CME strongly recommends that you download the current templates file every Sunday prior to market open.

Historical templates.xml files will be maintained on the ftp site and stored in a directory indicating the dates they were effective. The historical templates will be moved to the "Archive" directory for the corresponding environment.

Template Versioning

The templates.xml file is versioned each time an update is made. You can determine whether the client system has the latest templates or if the new templates.xml file needs to be downloaded and applied.

The version number changes whenever there is a change to the template, such as an:

*addition/removal/modification of a FIX tag
*addition/removal/modification of a FIX tag value
*addition of a new template

See also:

- Template Overview - XML Template Example
- Template Overview - Implementation Considerations
- Template Overview - Template Modification
- Market Data - Sample Template

Template Overview - XML Template Example

A template consists of Field Instructions that define the fields contained in the message. Field Instructions specify the field name, tag number, data type, field operator, and presence attribute that indicates if a field is optional or mandatory.

A sample market data template is shown below. The syntax is standard XML and can be parsed using a variety of open source tools. Valid template syntax is determined by the FAST Template Schema which is available in the FAST v1.1 specification.

*The real-time feed templates are based on the Market Data Incremental Refresh message type.

The information contained in a template is passed to the client FAST decoder at run-time such that the decoder recognizes how each field is encoded in terms of data type representation (Transfer) and data redundancy removal (Field).

The template is constructed of several sections including Template Identification, Header, Body and Sequence. Template Identification provides the template name and identifier. The Header includes FIX header fields such as ApplVerID (tag 1128), MsgType (tag 35), and SendingTime (tag 52). The Body provides information common across all repeating groups. Sequence represents a repeating group with a corresponding length field and a set of repeating group fields which carry the detailed entry information.

```
<?xml version="1.0" encoding="UTF-8"?>
<template name="MDIncRefresh" id="30" xmlns="http://www.cme.com/mdp/fast-templates" >
  <typeRef name="MDIncRefresh"/>
  <string name="ApplVerID" id="1128"> <constant value="7"/> </string>
  <string name="MessageType" id="35"> <constant value="X"/> </string>
  <string name="SenderCompID" id="49"> <constant value="CME"/> </string>
  <uInt32 name="MsgSeqNum" id="34"> </uInt32>
  <uInt64 name="SendingTime" id="52"> <delta/> </uInt64>
  <uInt32 name="TradeDate" id="75"><copy/> </uInt32>
  <sequence name="MDEntries"><length name="NoMDEntries" id="268"></length>
    <decimal name="MDEntryPx" id="270">
      <exponent><copy value="-2"/></exponent>
      <mantissa><delta/></mantissa> </decimal>
    <int32 name="MDEntrySize" id="271"> <delta/> </int32>
    <uInt32 name="MDEntryTime" id="273"> <delta/> </uInt32>
    <uInt32 name="MDPriceLevel" id="1023" presence="optional"> <increment/> </uInt32>
    <uInt32 name="MDUpdateAction" id="279"> <copy value="1"/> </uInt32>
    <string name="MDEntryType" id="269"> <copy value="0"/> </string>
    <uInt32 name="SecurityID" id="48"> <copy/> </uInt32>
    <uInt32 name="SecurityIDSource" id="22"> <constant value="8"/> </uInt32>
    <int32 name="NumberOfOrders" id="346" presence="optional"> <delta/> </int32>
    <string name="QuoteCondition" id="276" presence="optional"> <copy value="K"/> </string>
    <string name="TickDirection" id="274" presence="optional"> <default/> </string>
    <int32 name="NetChgPrevDay" id="451" presence="optional"> <default/> </int32>
    <string name="TradeCondition" id="277" presence="optional"> <default/> </string>
    <int32 name="TradeVolume" id="1020" presence="optional"> <default/> </int32>
    <string name="TradingSessID" id="336"><default value="1"/> </string>
  </sequence>
</template>
```

Template Overview - Implementation Considerations

The following items should be considered before implementing template functionality:

- Client systems should use the defined sizes and type for each tag in the FIX Message Specification as a guide for storing data.

  > **Do not use the template to define this**.

- Any change to the template will result in an update to the template ID.
- The following template changes should be handled by the client without any changes to their decoder.

> Customers will be notified prior to a template change via a Globex Notice and a Market Data Notice.

Template Change	Decoder Impact	Template Release Plan	Additional Information

New Tags (CME Group originated)	none	A template that contains a new tag (CME Group originated) will be released in the New Release Certification environment approximately 4 weeks prior to Production environment (and Certification environment) roll-out.	A new tag may be defined as a result of a change in the business logic. This change should not require a modification to the client system decoder (however) there may be changes required to the clients back-end systems if they choose to implement this change.
New Tags (FIX originated)	none	A template that contains a new tag (FIX originated) will be released in the New Release environment approximately 4 weeks prior to Production environment (and Certification environment) roll-out.	A new tag may be defined as a result of a change in the business logic. This change should not require a modification to the client system decoder (however) there may be changes required to the clients back-end systems if they choose to implement this change.
Removed Tags	none	A template in which a tag is no longer available will be released in the New Release environment approximately 4 weeks prior to Production environment (and Certification environment) roll-out.	A tag may be removed (for example) if it is no longer needed.
Modified Tags	none	A template that contains a modified tag will be released in the New Release environment approximately 4 weeks prior to Production environment (and Certification environment) roll-out.	A modification to an existing tag may be defined as a result of a change in the business logic. This tag modification should not require a modification to the client system decoder (however) there may be changes required to the clients back-end systems if they choose to implement this change.
Rename or Reorder Tags	none	A template that contains a renamed or reordered tag will be released in the New Release environment approximately 2 weeks prior to Production environment (and Certification environment) roll-out.	A renamed or reordered tag may be defined to increased efficiency. This tag modification should not require a modification to the client system decoder (however) there may be changes required to the clients back-end systems if they choose to implement this change.
New Message Types	none	A template that contains a new message type will be released in the New Release environment approximately 4 weeks prior to Production environment roll-out.	A new message type may be defined as a result of enhanced CME Globex functionality (or a change in business logic). This change should not require a modification to the client system decoder (however) there may be changes required to the clients back-end systems if they choose to implement this change.

Modified Operators	none	A template that contains a modified operator will be released in the New Release environment approximately 2 weeks prior to Production environment roll-out.	
Modified Data Types	none	A template that contains a modified data type will be released in the New Release environment approximately 2 weeks prior to Production environment roll-out.	

Template Overview - Template Modification

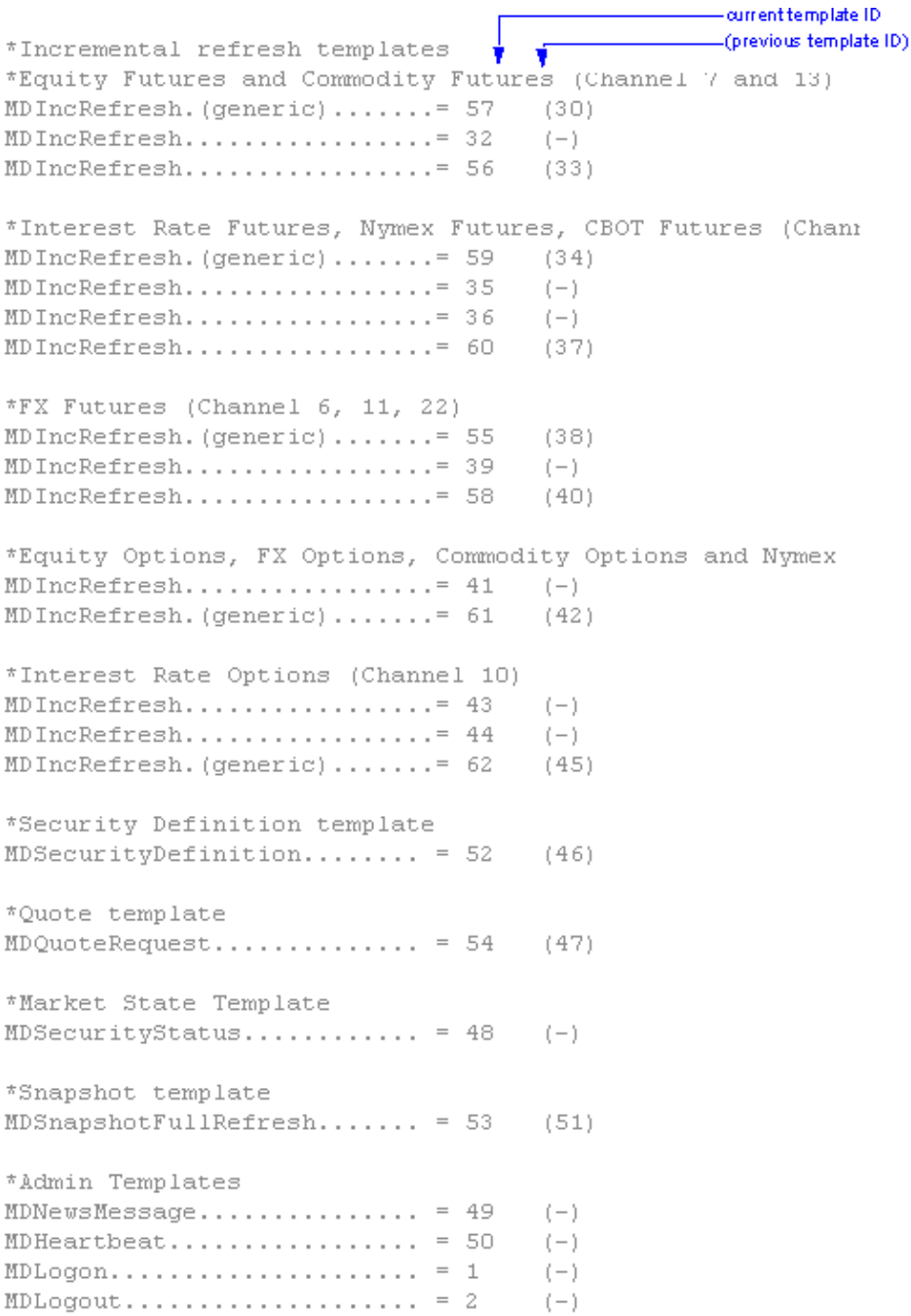
A template archiving process is available. From the template ftp site, the 'Templates' directory for the corresponding environment (Production, New Release, and Certification) contains two sub-directories:

- **Active** - contains the current templates.xml file.
- **Archive** - contains all retired versions of the templates.xml file.

When modifying the templates.xml file, CME Group makes a copy of the active version, adds the next available template identifier, then makes the necessary changes in the templates.xml file and replaces it in the Active directory. Two versions of the template remain in the templates.xml file, with the previous template listed under the new template. When a template is changed for the third time, the oldest version of the template is moved to the templates.xml file in the Archive directory.

Example:
If template 30 previously replaced template 28 and now a new template 57 is being generated to replace template 30, then template 28 is removed from the templates.xml Active directory and added to the Archive directory. Templates 57 and 30 remain in the Active directory.

The template.xml file in the Active directory lists both the current and the previous template IDs, as displayed in the following example:



The following example displays the current template name "57" with the previous template name "30" listed below it. A note below the previous template indicates the current template ID.

```
  -->
+ <template name="MDIncRefresh_57" id="57" dictionary="57" xmlns="http://www.fixprotocol.org/ns/fast/td/1.1">
- <template name="MDIncRefresh_30" id="30" dictionary="30" xmlns="http://www.fixprotocol.org/ns/fast/td/1.1">
    <!-- desc="DEPRECATED. USE VERSION 57"  -->
  - <string name="ApplVerID" id="1128">
      <constant value="8" />
    </string>
```

Market Data - Sample Template

Line #	Template Syntax	Use and Description
2	<template name="MDIncRefresh" id="35">	Provides the template name (MDIncRefresh) and template identifier (35).
3	<typeRef name="MDIncRefresh" />	Specifies the type reference name that allows the template to be referenced in other templates.
4	<string name="ApplVerID" id="1128"> <constant value="FIX.5.0" /> </string>	Field instruction for ApplVerID defined as a string with an identifier of 1128 corresponding to the FIX tag number. ApplVerID has a constant field operator with a value of FIX.5.0 indicating the FIX version.
5	<string name="MessageType" id="35"> <constant value="X" /> </string>	Field instruction for MessageType defined as a string with identifier = 35 corresponding to the FIX tag number. MessageType has a constant field operator with a value of X which indicates the FIX message type---in this case Market Data Incremental Refresh.
6	<string name="SenderCompID" id="49"> <constant value="CME" /> </string>	Field instruction for SenderCompID defined as a string with identifier = 49 corresponding to the FIX tag number. SenderCompID has a constant field operator with a value of 'CME' which indicates the originator of the data.
7	<uInt32 name="MsgSeqNum" id="34"> <increment /> </uInt32>	Field instruction for MsgSeqNum defined as an unsigned integer with identifier = 34 corresponding to the FIX tag number. MsgSeqNum has an increment field operator.
8	<uInt64 name="SendingTime" id="52"> <delta /> </uInt64>	Field instruction for SendingTime defined as an unsigned integer and with identifier = 52 corresponding to the FIX tag number. SendingTime has a delta field operator.
9	<sequence name="MDEntries"> <length name="NoMDEntries" id="268"> <copy /> </length>	Sequence instruction demarks the beginning of the MDEntries repeating group. The sequence includes a length field called 'NoMDEntries' that specifies the number of repeating groups present in the message. NoMDEntries has a copy field operator

10	<int32 name="MDEntryPx" id="270"> <delta /> </int32>	Field instruction for MDEntryPx (first field instruction in repeating group) defined as a signed integer with identifier = 270 corresponding to the FIX tag number. MDEntryPx has a delta field operator.
11	<int32 name="MDEntrySize" id="271"> <delta /> </int32>	Field instruction for MDEntrySize defined as a signed integer with identifier = 271 corresponding to the FIX tag number. MDEntrySize has a delta field operator
12	<uInt32 name="MDEntryTime" id="273"> <delta /> </uInt32>	Field instruction for MDEntryTime which is defined as an unsigned integer with identifier = 273 corresponding to the FIX tag number. MDEntryTime has a delta field operator.
13	<uInt32 name="MDPriceLevel" id="1023" presence="optional"> <increment /> </uInt32>	Field instruction for MDPriceLevel defined as an unsigned integer with identifier = 1023 corresponding to the FIX tag number. MDPriceLevel has a delta field operator.
14	<uInt32 name="MDUpdateAction" id="279"> <copy value="0" /> </uInt32>	Field instruction for MDUpdateAction defined as an unsigned integer and identifier = 279 corresponding to the FIX tag number. MDUpdateAction has a copy field operator with a value of 0.
15	<string name="MDEntryType" id="269"> <default value="1" /> </string>	Field instruction for MDEntryType which is defined as a string and has an identifier of 269 which corresponds to the FIX tag number. MDEntryType has a default field operator with a value of 1.
16	<int32 name="SecurityID" id="48"><delta/></int32>	Field instruction for SecurityID defined as an unsigned integer and has an identifier = 48 corresponding to the FIX tag number. SecurityID has a delta field operator.
16	<uint32 name="SecurityID" idSource id="22"><constant value="8"></uint32>	Field instruction for SecurityIDSource defined as an unsigned integer and has an identifier = 22 corresponding to the FIX tag number. SecurityDesc has a constant field operator.
17	<int32 name="NumberOfOrders" id="346" presence="optional"> <delta /> </int32>	Field instruction for NumberOfOrders defined as a signed integer and with identifier = 346 corresponding to the FIX tag number. NumberOfOrders has a delta field operator.
18	<string name="QuoteCondition" id="276" presence="optional"> <copy /> </string>	Field instruction for QuoteCondition defined as a string with identifier = 276 correspondsing to the FIX tag number. QuoteCondition has a copy field operator.

19	<string name="TickDirection" id="274" presence="optional"> <default /> </string>	Field instruction for TickDirection is defined as a string with identifier = 274 corresponding to the FIX tag number. TickDirection has a default field operator.
20	<uint32 name="NetChgPrevDay" id="451" presence="optional"> <default /> </uint32>	Field instruction for NetChgPrevDay defined as an unsigned integer with identifier = 451 corresponding to the FIX tag number. NetChgPrevDay has a default field operator.
21	<string name="TradeCondition" id="277" presence="optional"> <default /> </string>	Field instruction for TradeCondition defined as a string with identifier = 277 corresponding to the FIX tag number. TradeCondition has a default field operator.
22	<int32 name="TradeVolume" id="1020" presence="optional"> <default /> </int32>	Field instruction for TradeVolume defined as a signed integer with identifier = 1020 corresponding to the FIX tag number. TradeVolume has a default field operator.
23	<string name="TradingSessID" id="336"> <default value="1" /> </string>	Field instruction for TradingSessID defined as a string with identifier = 336 corresponding to the FIX tag number. TradingSessID has a default field operator. TradingSessID is the last field in the Sequence

Market Data - Incremental Book Management

CME Group uses an electronic market data format based on the FIX standard which minimizes the amount overall content of data transmitted through an incremental book management approach. FIX provides a flexible protocol for market data messaging which is well suited for transmitting electronic books from the CME Globex trading platform to customers. Incremental book management provides significant efficiency and flexibility across the entire market data infrastructure. The use of incremental FIX market data messaging in combination with FAST compression produces a highly optimized feed which results in bandwidth savings as well as latency reductions.

CME Group represents markets in executable orders and quotes using a central limit order book. This order book is constantly changing as market events cause orders to be added, modified, and canceled. Updates are then sent out over a market data stream so client systems can then construct a copy of the book in order to track prices in the market and submit appropriately priced orders.

Multiple data blocks may be sent in the same message. The number of data blocks within the message is indicated in tag 268-NoMDEntries. The message can contain data blocks for different instruments or entry types (book update, statistics, trades, etc).

FIX Message Structures

The Market Data Incremental Refresh (tag 35-MsgType = X) message is used to apply instructions to a book. These instructions are incremental and update applicable parts of the book as

necessary, as opposed to refreshing the entire book each time there is an update. This message is used to maintain the aggregate order book for CME Group products. This message, on a real-time basis, is also used to send statistics.

The FIX Market Data message has a FIX header followed by a number of data blocks. Each data block represents a single instruction such as a book update or trade. This section refers to the items as data blocks when describing how the book is maintained. A single FIX Market Data message can contain many data blocks, such as New bid/ask, Change bid/ask, Delete bid/ask, and Overlay bid/ask, across multiple instruments. A book is not valid until all book data blocks for an instrument are processed in a Market Data Incremental Refresh (tag 35-MsgType =X) message.

Common Incremental Refresh Book Update Tags

Tag	Field Name	Description
279	MDUpdateAction	Type of market data update action (e.g. Change)
269	MDEntryType	Type of market data entry (e.g. Bid)
83	RptSeq	Sequence number per instrument update
276	QuoteCondition	Describes quote (e.g. implied)
48	SecurityID	Unique instrument ID as qualified by the exchange in tag 22-SecurityIDSource
22	SecurityIDSource	Identifies source of tag 48-SecurityID value. This value is always '8' for CMEG and is required if tag 48-SecurityID is specified.
1023	MDPriceLevel	Position in the book.
273	MDEntryTime	Time of market data entry.
270	MDEntryPx	Price of market data entry.
271	MDEntrySize	Quantity or volume represented by the market data entry.
346	NumberOfOrders	Number of orders in the market (market by order) or at the given price level (market by price)
1070	MDQuoteType	Identifies quote type (e.g. tradable)

See also:

- Market Data - Central Limit Order Book
- Market Data - Multiple Depth Book
- Market Data - Implied Book
- Market Data - Consolidating Implied and Multiple Depth Books into a Single Book
- Market Data - Top of Book
- Market Data - CME Globex Pricing

Market Data - Central Limit Order Book

The central limit order book is available over a market data feed that provides an aggregate book with a pre-determined number of price levels. The term "aggregate book" is used to refer to the case where orders are summarized at each price level. The quantity associated with that price level is the aggregation of all individual order quantities for that price level. CME Group offers the following:

- multiple-depth book (3, 5, 10)
- top-of-book
- 2-deep book for implied futures price

- Client systems must determine the book-depth for an instrument from tag 264-MarketDepth in the Security Definition (tag 35-MsgType=d) message.
- Client systems must be able to process all valid values in tag 279-MDUpdateAction.

It is possible to deliver market data for instrument order books with different book depths on a single channel, for better bandwidth efficiency. For example, on Channel 7, Instrument A can have a 5-deep order book, while Instrument B has a 10-deep order book. Tag 264-MarketDepth in the Security Definition (Tag 35-MsgType=d) message must be referenced to obtain the book depth per instrument.

Example - Multiple Book Depths

Market Data Channel	Instrument	Book Depth
Channel 7	Instrument x	5
Channel 7	Instrument y	10

Market Data - Multiple Depth Book

The examples in this topic illustrate the mechanics of a 5-deep book. All books that are not top-of-book will use the same mechanics.

CME Group provides a multiple-depth book for most products. Client systems must determine the book depth for an instrument from tag 264-MarketDepth in the Security Definition (tag 35-MsgType=d) message. The aggregate book reports summarized order quantities and order counts at a given price level. The depth represents the number of price levels that are supported in the feed. The view can be represented as a number of rows in a table for each of the bid and ask sides. On each side, there are a number of rows showing the quantity available at a number of price levels. An aggregate depth book is sequenced by price, descending for bid and ascending for ask.

5-Deep - Best Bid/Ask

Bid			Ask		
Order Count	Quantity	Price	Price	Quantity	Order Count
1	100	9427.5	9428	40	2
19	500	9427	9428.5	600	35
34	750	9426.5	9429	850	55
25	400	9426	9429.5	350	21
14	300	9425.5	9430	150	12

CME Group maintains the Aggregate Depth view with the following data blocks:

- Add - to create/insert a new price at a specified price level (tag 279 MDUpdateAction=0)
- Change - change quantity for a price at a specified price level (tag 279 MDUpdateAction=1)
- Delete - remove a price at a specified price level (tag 279 MDUpdateAction=2)

An Aggregate book is built from a series of data blocks which indicate whether an entry is to be inserted (Add), changed (Change), or removed (Delete). All data blocks are issued for a specified entry type (tag 269), price (tag 270), and price level (tag 1023). The incremental instruction approach assumes the use of the Market Data Incremental Refresh (tag 35-MsgType =X) message. The Bid and Ask sides are updated independently with separate data blocks. The practice of sending separate data blocks provides efficiencies by allowing only the bid or ask to be sent, based on which side has changed, rather than both sides.

CME Globex sends an add data block if there is a new price level. Client systems should then shift price levels down, and delete any price levels past the defined depth of the book as indicated in tag 264-MarketDepth in the Security Definition (tag 35-MsgType=d) Message.

CME Globex sends a delete data block to remove a price level in the book. Client systems should shift prices below the data block up to the price level vacated by the deleted price level. If available, an add data block will be sent to fill in the last price level.

The change data block is sent to update characteristics of a price level without changing the price itself, or impacting any other prices on the book. The change data block is sent to update the order count and / or quantity for a price level. The change data block is not sent when the price changes at a given price level.

In general, if a trade occurs, CME Group will send a delete or change data block to update the book. The trade data block itself is not used to update the order book.

> ⓘ The last best price data block will appear similar to a book update data block. To identify that the data block is a last best price data block (and NOT a book update), verify that tag 276-QuoteCondition contains a value of C (Exchange Best).

Examples

Market Data Book Update - Modify Bid Quantity
Market Data Book Update - Order Cancelled and Replaced
Market Data Book Update - New Order Entered at Same Price
Market Data Book Update - New Best Price Entered

Market Data Book Update - Modify Bid Quantity

This example shows the update performed on an existing book when a Modify Bid Quantity data block is processed. Quantity represents aggregate quantity for all orders at the given price level and order count and is the number of orders at the given price level.
Starting Book

Bid			Ask		
Order Count	Quantity	Price	Price	Quantity	Order Count
1	100	9427.5	9428	40	2
19	500	9427	9428.5	600	35
34	750	9426.5	9429	850	55
25	400	9426	9429.5	350	21

14	300	9425.5	9430	150	12

Modify Bid Quantity (tag 35-MsgType=X) Data Block

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	1	1 = change. Type of Market Data update action.
269	MDEntryType	0	0 = bid. Type of Market Data entry.
83	RptSeq	e.g., 123	Sequence number per Instrument update.
1023	MDPriceLevel	1	Position in the book.
273	MDEntryTime	e.g., 22:43.19.000	Time of Market Data Entry.
271	MDEntrySize	90	Quantity represented by the Market Data Entry.
270	MDEntryPx	9427.50	Price of the Market Data Entry.
346	NumberOfOrders	1	Number of orders at price level.
48	SecurityID	given	Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	8 = CME. Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

Updated Book

Bid			Ask		
Order Count	Quantity	Price	Price	Quantity	Order Count
1	90	9427.5	9428	40	2
19	500	9427	9428.5	600	35
34	750	9426.5	9429	850	55
25	400	9426	9429.5	350	21
14	300	9425.5	9430	150	12

Market Data Book Update - Order Cancelled and Replaced

In this example, the canceled order is the only order at the given price level. Therefore, the first data block for the instrument removes the price level from the book for the canceled order, and the subsequent data block updates the book with the new order. In this example, the book shows

the removal of price level 1, followed by an addition to price level 5.

> Client systems should shift prices below the data block up to the price level vacated by the deleted price level. If all levels in the book are full, an add data block will be sent to fill in the last price level.

Bid			Ask		
Order Count	Quantity	Price	Price	Quantity	Order Count
1	90	9427.5	9428	40	2
19	500	9427	9428.5	600	35
34	750	9426.5	9429	850	55
25	400	9426	9429.5	350	21
14	300	9425.5	9430	150	12

Book Update - Delete35=X Data Block

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	2	2 = delete. Type of Market Data update action.
269	MDEntryType	0	0 = bid. Type of Market Data entry.
83	RptSeq	e.g. 124	Sequence number per Instrument update.
1023	MDPriceLevel	1	Position in the book
273	MDEntryTime	e.g. 09:23:12:32:2012	Time of Market Data Entry.
271	MDEntrySize	90	Quantity represented by the Market Data Entry.
270	MDEntryPx	9427.50	Price of the Market Data Entry.
346	NumberOfOrders	1	Number of orders at price level.
48	SecurityID	given	Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	8 = CME. Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

Updated Book

Bid			Ask		
Order Count	Quantity	Price	Price	Quantity	Order Count

19	500	9427	9428.5	600	35
34	750	9426.5	9429	850	55
25	400	9426	9429.5	350	21
14	300	9425.5	9430	150	12

Book Update - AddStarting Book35=X Data Block

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = add. Type of Market Data update action.
269	MDEntryType	0	0 = bid. Type of Market Data entry.
83	RptSeq	e.g. 125	Sequence number per Instrument update.
1023	MDPriceLevel	5	Position in the book
273	MDEntryTime	e.g. 09:23:12:32:2015	Time of Market Data Entry.
271	MDEntrySize	400	Quantity represented by the Market Data Entry.
270	MDEntryPx	9425.00	Price of the Market Data Entry.
346	NumberOfOrders	1	Number of orders at price level.
48	SecurityID	given	Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	8 = CME. Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

Updated Book

Bid			Ask		
Order Count	Quantity	Price	Price	Quantity	Order Count
19	500	9427	9428	40	2
34	750	9426.5	9428.5	600	35
25	400	9426	9429	850	55
14	300	9425.5	9429.5	350	21
1	400	9425	9430	150	12

Market Data Book Update - New Order Entered at Same Price

This example shows the book update that occurs for a new order submitted at an existing price level.

Bid			Ask		
Order Count	Quantity	Price	Price	Quantity	Order Count
19	500	9427	9428	40	2
34	750	9426.5	9428.5	600	35
25	400	9426	9429	850	55
14	300	9425.5	9429.5	350	21
1	400	9425	9430	150	12

35=X Data Block

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	1	1 = change. Type of Market Data update action.
269	MDEntryType	0	0 = bid. Type of Market Data entry.
83	RptSeq	e.g. 127	Sequence number per Instrument update.
1023	MDPriceLevel	1	Position in the book.
273	MDEntryTime	e.g. 09:23:12:32:2019	Time of Market Data Entry.
271	MDEntrySize	503	Quantity represented by the Market Data Entry.
270	MDEntryPx	9427.00	Price of the Market Data Entry.
346	NumberOfOrders	20	Number of orders at price level.
48	SecurityID	given	Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	8 = CME. Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

Updated Book

Bid			Ask		
Order Count	Quantity	Price	Price	Quantity	Order Count

20	503	9427	9428	40	2
34	750	9426.5	9428.5	600	35
25	400	9426	9429	850	55
14	300	9425.5	9429.5	350	21
1	400	9425	9430	150	12

Market Data Book Update - New Best Price Entered

This example shows an order entered at a new best price level.

Bid			Ask		
Order Count	Quantity	Price	Price	Quantity	Order Count
20	503	9427	9428	40	2
34	750	9426.5	9428.5	600	35
25	400	9426	9429	850	55
14	300	9425.5	9429.5	350	21
1	400	9425	9430	150	12

35=X Data Block

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	0	0 = bid. Type of Market Data entry.
83	RptSeq	e.g. 131	Sequence number per Instrument update.
1023	MDPriceLevel	1	Position in the book.
273	MDEntryTime	e.g. 09:23:12:32:2021	Time of Market Data Entry.
271	MDEntrySize	200	Quantity represented by the Market Data Entry.
270	MDEntryPx	9427.50	Price of the Market Data Entry.
346	NumberOfOrders	1	Number of orders at price level.
48	SecurityID	given	Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.

22	SecurityIDSource	8	8 = CME. Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

Updated Book

Bid			Ask		
Order Count	Quantity	Price	Price	Quantity	Order Count
1	200	9427.5	9428	40	2
20	503	9427	9428.5	600	35
34	750	9426.5	9429	850	55
25	400	9426	9429.5	350	21
14	300	9425.5	9430	150	12

Market Data - Implied Book

CME Group provides a 2-deep best bid and ask in the market for each implied prices futures contract. Implied book updates are denoted by the presence of tag 276-QuoteCondition = K (Implied). Trade details and instrument status are provided in separate data blocks

CME Group maintains the Aggregate Depth view with the following data blocks:

- **Add** - to create/insert a new price at a specified price level (tag 279-MDUpdateAction=0)
- **Change** - change quantity of a price at a specified price level (tag 279-MDUpdateAction=1)
- **Delete** - remove a price at a specified price level (tag 279-MDUpdateAction=2)

An Aggregate book is built from a series of data blocks which indicate whether an entry is to be inserted (Add), changed (Change), or removed (Delete). All data blocks are issued for a specified entry type (tag 269-MDEntryType), price (tag 270-MDEntryPx), price level (tag 1023-MDPriceLevel), and quantity (tag 271-MDEntrySize). The incremental instruction approach assumes the use of the Market Data Incremental Refresh message (tag 35-MsgType=X). The Bid and Ask sides are updated independently with separate data blocks. The practice of sending separate data blocks provides efficiencies by allowing only the bid or ask to be sent, based on which side has changed, rather than both sides.

CME Globex sends an add data block if there is a new price level. Client systems should then shift price levels down, and delete any price levels past 2-deep.

CME Globex sends a delete data block to remove a price level in the book. Client systems should shift prices below the data block up to the price level vacated by the deleted price level. If all levels in the book are full, an add data block will be sent to fill in the last price level.

The change data block is sent to update characteristics of a price level without changing the price itself, or impacting any other prices on the book. The change data block is sent to update the order quantity for a price level. The change data block is not sent when the price changes at a given price level.

If a trade occurs, CME Group will send a delete or change data block to update the book. The trade data block itself is not used to update the order book.

Examples

This example shows how a book is built and updated. The book generally consists of a set of bid prices and ask prices, in which bids are descending and asks are ascending. The quantity is provided at each price level. For implied prices futures, a two-deep book is available.
The following table illustrates an initial implied market book.
2-Deep Book

Bid		Ask	
Quantity	Price	Price	Quantity
100	9427.50	9428.00	40
200	9427.00	9428.50	100

- Implied Book - Basic Book Update Data Block
- Implied Book - Bid Side Quantity Modified
- Implied Book - Price Level Delete and New Price Level
- Implied Book - New Order Entered at Existing Price Level

Implied Book - Basic Book Update Data Block

A FIX message sent to update the top of book will update one side only. The tags normally sent for a book
update data block are:

> tag 346-NumberOfOrders is not included in this data block.

- tag 279-MDUpdateAction
- tag 269-MDEntryType
- tag 83-RptSeq
- tag 276-QuoteCondition
- tag 1023-MDPriceLevel
- tag 273-MDEntryTime
- tag 271-MDEntrySize
- tag 270-MDEntryPx
- tag 48-SecurityID
- tag 22-SecurityIDSource
- tag 336-TradingSessionID

Implied Book - Bid Side Quantity Modified

The quantity of an order can be modified. The book will show an update to the size displayed:
Bid 90 @ 9427.50
Book Update Instruction
Book Update - Data Block

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	1	1 = Change. Type of Market Data update action.
269	MDEntryType	0	0 = Bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
276	QuoteCondition		Space-delimited list of conditions describing a quote.

1023	MDPriceLevel	1	Price Level 1. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	90	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9427.50	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Updated Book

Bid		Ask	
Quantity	Price	Price	Quantity
90	9427.50	9428.00	40
200	9427.00	9428.50	100

Implied Book - Price Level Delete and New Price Level

This example shows a scenario in which a single order comprises the total quantity at the given price level and which is subsequently canceled and replaced. Resulting market data include a Delete instruction to remove the price level, and an New update to populate the price level.

> Client systems should shift prices below the data block up to the price level vacated by the deleted price level. If all levels in the book are full, an add data block will be sent to fill in the last price level.

Cancel bid 200 @ 9427.00
Book Update Instruction
Book Update - Data Block 1

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	2	2 = Delete. Type of Market Data update action.
269	MDEntryType	0	0 = Bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
276	QuoteCondition		Space-delimited list of conditions describing a quote.
1023	MDPriceLevel	2	Price Level 2. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	200	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9427.00	Price of the Market Data Entry.

48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Add bid 80 @ 9426.50
Book Update Instruction
Book Update - Data Block 2

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = Add. Type of Market Data update action.
269	MDEntryType	0	0 = Bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
276	QuoteCondition		Space-delimited list of conditions describing a quote.
1023	MDPriceLevel	2	Price Level 2. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	80	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9426.50	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Updated Book

Bid		Ask	
Quantity	Price	Price	Quantity
90	9427.50	9428.00	40
80	9426.50	9428.50	100

Implied Book - New Order Entered at Existing Price Level

When a new order is entered in the book at an existing price level, the book will update as follows:
New bid order 3 @ 9427.50
Book Update Instruction
Book Update - Data Block 1

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	1	1 = Change. Type of Market Data update action.
269	MDEntryType	0	0 = Bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
276	QuoteCondition		Space-delimited list of conditions describing a quote.
1023	MDPriceLevel	1	Price Level 1. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	93	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9427.50	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Updated Book

Bid		Ask	
Quantity	Price	Price	Quantity
93	9427.50	9428.00	40
80	9426.50	9428.50	100

Market Data - Consolidating Implied and Multiple Depth Books into a Single Book

The data block for a multiple-depth book reports modifications to prices, quantities, and order count, and the data block for an implied prices book reports modifications to prices and quantities. The multiple-depth book should be used in conjunction with implied prices book to create an accurate book for all contracts with implied functionality. To create a consolidated book, the multiple-depth book and the implied book must be built and managed separately, then consolidated to reflect the current state of the market.

Multiple-Depth Book

Bid			Ask		
Order Count	Quantity	Price	Price	Quantity	Order Count
19	500	9427.00	9428.00	40	2
34	750	9426.50	9428.50	600	35

25	400	9426.00	9429.00	850	55
14	300	9425.50	9429.50	350	21
10	200	9425.00	9430.50	150	12

Implied Book

Bid		Ask	
Quantity	Price	Price	Quantity
100	9427.50	9428.00	40
200	9427.00	9430.00	100

After building the multiple-depth book and implied books, the actual book can be built by merging the multiple-depth book and implied book tables.

Order count is not displayed in the combined book.

Consolidated Book

Bid		Ask	
Quantity	Price	Price	Quantity
100	9427.50	9428.00	80
700	9427.00	9428.50	600
750	9426.50	9429.00	850
400	9426.00	9429.50	350
300	9425.50	9430.00	100

Market Data - Top of Book

CME Group provides the current best bid and ask (top-of-book) in the market for certain instruments. Client systems must determine the book depth for an instrument from tag 264-MarketDepth in the Security Definition (tag 35-MsgType=d) message.

Trade details and instrument status are provided in separate data blocks. The following table illustrates the types of information that can be displayed:

1-deep Book - Best Bid/Ask

	Bid			Ask	
Order Count	Quantity	Price	Price	Quantity	Order Count
1	100	9427.50	9428.00	40	2

CME Group maintains a top-of-book view with the following data block:

- **Overlay** - add, modify, or delete an entry in the book (tag 279 MDUpdateAction=5).

When the best price changes, CME Globex sends a single overlay instruction for price level 1. Overlay is only used for instruments with order books that are one level deep.

Basic Book Update Data Block

A FIX message is sent to update a top-of-book updates one side only. The tags normally sent for a top-of-book data block are:

- tag 279-MDUpdateAction
- tag 269-MDEntryType
- tag 83-RptSeq
- tag 1023-MDPriceLevel
- tag 273-MDEntryTime
- tag 271-MDEntrySize
- tag 270-MDEntryPx
- tag 346-NumberOfOrders
- tag 48-SecurityID
- tag 22-SecurityIDSource
- tag 336-TradingSessionID

Indexing the Entry

Top-of-book entries are referenced using a composite index which consists of the tag 48-SecurityID, tag 269-MDEntryType (bid/ask) and tag 270-MDEntryPx. This set of fields acts as a composite key that allows the entry (bid or ask) to be accessed and subsequently, updated or deleted.

- Market Data - Top of Book - Examples

Market Data - Top of Book - Examples

- Quantity on Bid Side Modified
- Entire Order Canceled
- New Order Entered
- New Order Entered at Same Price

This example shows how a book is built and updated. The quantity and order count are provided.

The following table illustrates an initial market book.

Top-of-Book - Best Bid/Ask

Bid			Ask		
Order Count	Quantity	Price	Price	Quantity	Order Count
1	100	9427.50	9428.50	40	2

Quantity on Bid Side Modified

The quantity of an order can be modified. The book will show an update to the size displayed:
Bid 90 @ 9427.50
Book Update Instruction

Book Update - Data Block 1

Tag Number	Tag Name	Value	Description

279	MDUpdateAction	5	5 = overlay. Type of Market Data update action.
269	MDEntryType	0	0 = bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
1023	MDPriceLevel	1	Price level 1. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	90	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9427.50	Price of the Market Data Entry.
346	NumberOfOrders	1	Number of orders in the market.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Top-of-Book - Best Bid/Ask

Bid				Ask	
Order Count	Quantity	Price	Price	Quantity	Order Count
1	90	9427.50	9428.50	40	2

Entire Order Canceled

An entire order can be canceled, which removes the top of book. The book will show an update:
Cancel bid 90 @ 9427.50
Book Update Instruction

Book Update - Data Block 1

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	5	5 = overlay. Type of Market Data update action.
269	MDEntryType	0	0 =bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
1023	MDPriceLevel	1	Price level 1. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	0	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx		Price of the Market Data Entry.

346	NumberOfOrders	0	Number of orders in the market.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Top-of-Book - Best Bid/Ask

Bid			Ask		
Order Count	Quantity	Price	Price	Quantity	Order Count
			9428.50	40	2

New Order Entered

A new order can be entered to a book. The book will show an update:
New bid order 10 @ 9428.00
Book Update Instruction

Book Update - Data Block 1

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	5	5 = overlay. Type of Market Data update action.
269	MDEntryType	0	0 =bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
1023	MDPriceLevel	1	Price level 1. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	10	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9428.00	Price of the Market Data Entry.
346	NumberOfOrders	1	Number of orders in the market.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Top-of-Book - Best Bid/Ask

Bid			Ask		
Order Count	Quantity	Price	Price	Quantity	Order Count
1	10	9428.00	9428.50	40	2

New Order Entered at Same Price

A new order can be entered to a book at the same price level. The book will show an update:
New bid order 3 @ 9428.00
Book Update Instruction

Book Update - Data Block 1

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	5	5 = overlay. Type of Market Data update action.
269	MDEntryType	0	0 = bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
1023	MDPriceLevel	1	Price level 1. Position in the book.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	13	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9428.00	Price of the Market Data Entry.
346	NumberOfOrders	2	Number of orders in the market.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource		Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
336	TradingSessionID		Identifier for Trading Session.

Top-of-Book - Best Bid/Ask

Bid			Ask		
Order Count	Quantity	Price	Price	Quantity	Order Count
2	13	9428.00	9428.50	40	2

Market Data - CME Globex Pricing

This topic describes the pricing and tick convention used for instruments available on CME Globex. The CME Globex price is the format used in all price tags in order entry iLink and FIX/FAST market data interfaces. Pricing on CME Globex involves the following:

Tick Size Calculation

- Variable Tick Instrument
- Variable Tick Table
- Standard Tick Instrument

Suggested Display Format for Non-Fractional Products

Fractional Pricing

- Application of Display Factor
- Fractional Pricing - Tick Conversion Grid

Formatted Fractional Tick to Decimal

Tick Size Calculation

The tick is the minimum price fluctuation allowed for a futures or options contract during a trading session as specified by the contract terms of CME Group. An instrument can have either a variable tick (primarily for options instruments) or a standard tick. Both the variable tick and standard tick are obtained from the Security Definition (tag 35-MsgType=d).

See also:
- Variable Tick Instrument
- Variable Tick Table
- Standard Tick Instrument

Variable Tick Instrument

When tag 871-InstAttribType=23 in the Security Definition (tag 35-MsgType=d) message, the instrument is VTT, Variable Tick Table, eligible. The following process should be performed to calculate the tick size for a VTT eligible instrument. Tag 872-InstAttribValue will contain the VTT index code as shown in the Variable Tick Table.

> Tag 969-MinPriceIncrement will not be sent in the Security Definition (tag 35-MsgType=d) message for VTT-eligible instruments.

Example - Variable Tick Instrument

CME Globex sends the following Security Definition (tag 35-MsgType=d) message for an instrument with a
price of 510:

1128=9|35=d|49=CME|34=9647|52=20080608171603359|15=USD|22=8|48=300998|55=OS|107
P1750|200=200806|202=175000|207=XCME|461=OPAFPS|462=5|562=1|711=1|311=N/A|305=8|

827=0|864=2|865=5|866=20060707|1145=202500000|865=7|866=20080619|1145=202500000|
871=23|872=01|
871=24|872=1|871=24|872=2|871=24|872=3|871=24|872=14|947=USD|1140=9999|1141=1|10

1146=0|1147=0|1148=5|1149=9999900|1150=36530|1151=SP|1180=8|9787=0.01|9850=5

1. Tag 871-InstAttribType=23, so the instrument is VTT eligible instrument.
2. Tag 872-InstAttribValue contains the VTT index code 01.
3. Since tag 872-InstAttribValue is 01 for a market price of 510 (P = 510 > 500), the tick size for the contract is 10.

Variable Tick Table

Variable Tick Table Index Code	Current CME Globex Price	CME Globex Tick Size
01	P < -500	10
01	-500 ≤ P ≤ 500	5
01	P > 500	10
02	-5 ≤ P ≤ 5	0.5
02	P > 5	1
02	P < -5	1
03	-10 ≤ P ≤ 10	1
03	P > 10	2
03	P < -10	2
04	P < -500	25
04	-500 ≤ P ≤ 500	5
04	P > 500	25
10	P < -300	25
10	-300 ≤ P ≤ 300	5
10	P > 300	25
11	P < -300	10
11	-300 ≤ P ≤ 300	5
11	P > 300	10
12	P < -5	0.50
12	-5 ≤ P ≤ 5	0.25
12	P > 5	0.50

Standard Tick Instrument

If the Security Definition (tag 35-MsgType=d) message for an instrument does not contain tag 871-InstAttribType=23, the instrument uses a standard tick. The value found in tag 969-MinPriceIncrement is the tick size.
Example - Standard Tick Instrument

1128=9|9=425|35=d|49=CME|34=13591|52=20080522221850254|15=USD|22=8|48=806800|55

827=2|864=2|865=5|866=19980617|1145=213000000|865=7|866=20080616|1145=100000000|

871=24|872=5|947=USD|969=0.5

|1140=30000|1141=1|1022=GBX|264=5|1142=A|1143=10.00|1144=3|1146=12.5000|1148=9559

1151=GE|1180=9|9787=0.01|10=216

tick size = 0.5.

Suggested Display Format for Non-Fractional Products

The client system can use the tags shown in the following table to display the price of the instrument. The display factor is multiplied by both the CME Globex Tick and the CME Globex Price conventions to calculate the display tick and display price.

The following table is a sample display of CME Globex ticks and prices.

> The resulting display of price and tick below are suggested by CME Group but are not required.

Instrument	CME Globex Tick tag 969-MinPriceIncrement	CME Globex Price (tag 270-MDEntryPx)	Display Factor (tag 9787-DisplayFactor)	Display Tick	Display Price
ESH2	25	113700	0.01	.25	1137.00
GEM2	.5	9886.5	0.01	.005	98.865

CME Globex Price x Display Factor = Display Price
CME Globex Tick x Display Factor = Display Tick

Market Data - CME Globex Pricing Fractional Pricing

The information in this section is for products that require a decimal-to-fractional price conversion. These products are identified in the Security Definition (tag 35-MsgType=d) message by the following tags:

- tag 871-InstAttributeType='24'
- tag 872-InstAttribValue='12'
- Products that do not require a decimal-to-fractional price conversion will not contain tag 871-InstAttributeType='24' and tag 872-InstAttribValue='12'. For those products, customers can use the Display Factor (tag 9787-DisplayFactor) for FIX/FAST to determine delimiter placement.

> Display Factor (tag 9787-DisplayFactor) should not be used for fractionals.

MDP FIX/FAST Format

Fractional price data in FIX/FAST is transmitted in four repeating groups of tag 871-InstAttributeType and tag 872-InstAttribValue. The first repeating group indicates that a fractional display price is present, the second repeating group indicates main fraction denominator, the third repeating group indicates the sub fraction denominator, and the fourth repeating group indicates the number of decimals in the displayed price. These repeating groups function as follows:

Tag	FIX Name	Description
>871	InstAttributeType	'24' = Eligibility
>872	InstrAttribValue	'12' = When the preceding tag 871-InstAttributeType = '24' and this tag = '12' this indicates the product has a fractional price.

>871	InstAttributeType	'25' = Main Fraction
>872	InstrAttribValue	When the preceding tag 871-InstAttributeType = '25' this tag will contain the main fraction denominator.
>871	InstAttributeType	'26' = Sub Fraction
>872	InstrAttribValue	When the preceding tag 871-InstAttributeType = '26' this tag will contain the sub fraction denominator.
>871	InstAttributeType	'27' = Display Price Precision
>872	InstrAttribValue	When the preceding tag 871-InstAttributeType = '27' this tag will contain the number of decimals in the displayed price.

Example: ZNZ9 10 Year U.S. Treasury Note Future

```
1128=9|9=455|35=d|49=CME|34=523|52=20080525164027159|15=USD|22=8|48=50026

562=1|827=2|864=2|865=5|866=20070919|1145=213000000|865=7|866=20081219|114

871=24|872=6|871=24|872=7|
871=24|872=12|871=25|872=32|871=26|872=2|871=27|872=3
|947=USD|969=0.015625|1140=49999|1141=110|22=GBX|

264=5|1142=f|1143=0.937500|1146=0.001000|1150=113.718750|1151=ZN|1180=115|9
```

TDFT= Tick Display Format Type (Numerator and Denominator)

NDDP = Number of Decimals in the Display Price

10 Yr U.S. Treasury Note Future, TDFT Half 1/32 Tick, NDDP 3
Original decimal price 112.625

Fractional part after conversion 20 (.625 x 32)

Integer part and fractional part 112 20

Apply NDDP of 03

112 20

123 (Display places)

Resulting formatted price 112'200

Market Data - Real Time Statistics and Market Behavior Events

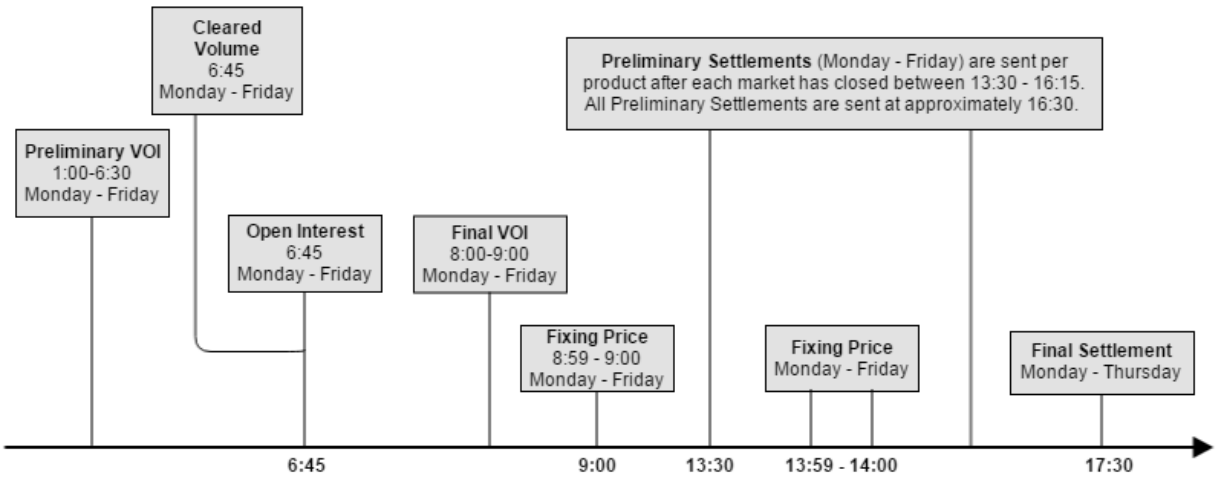
There are a number of statistics (market data events) which are related to changes in a book but are not used to update the book. The following artifacts fit this category: last best price, trade, high/low trade price, best high bid, and best low ask, price, and pre-opening statistics. These events describe the behavior of the market and allow a user to know when the market is moving in a certain direction and provide historical information on how the market has performed.

Multiple data blocks may be sent in the same message. To determine the number of data blocks you will receive in the message, refer to tag 268-NoMDEntries. Within the message, data blocks may be for different instruments or entry types (book update, statistics, or trades).

CME Globex Message Timeline

The following timeline shows when certain messages are typically sent on a given trading day.

> The following times are subject to change without notice



See also:

- Market Data - Pre-Opening Statistics
- Market Data - Last Best Price
- Market Data - Trade
- Market Data - Session High and Low Trade Price
- Market Data - High Bid and Low Ask
- Market Data - Open Interest
- Market Data - Cleared Volume
- Market Data - Fixing Price
- Market Data - Settlement Price
- Market Data - Trading Statistics Reset Based on Market State Value
- Market Data - Opening Traded Price

Market Data - Pre-Opening Statistics

Pre-opening statistics are used to indicate simulated buy or sell, stop spike, and pre-opening prices. Simulated buy provides the indicative best buy price that would be present after opening. Simulated sell provides the indicative best sell price that would be present after opening.

FIX Syntax for Indicative Opening - Market Data Incremental Refresh (tag 35-MsgType =X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	4	4 = opening price. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.

273	MDEntryTime		Time of Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
286	OpenCloseSettleFlag	5	Indicative opening price.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

FIX Syntax for Simulated Sell - Market Data Incremental Refresh (tag 35-MsgType =X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	E	E = simulated sell. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
271	MDEntrySize		Quantity or volume represented by the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

FIX Syntax for Simulated Buy - Market Data Incremental Refresh (tag 35-MsgType =X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	F	F = simulated buy. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
271	MDEntrySize		Quantity or volume represented by the Market Data Entry.

48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

Market Data - Last Best Price

The Last Best Price data block indicates that a new best bid or new best ask price has occurred, and implicitly conveys events which affect the top of the book. Last Best Price data blocks are denoted by the presence of tag 276-QuoteCondition = C (Exchange Best).

In certain cases, such as an order submission immediately followed by an order cancellation, the top of the book can be bettered without resulting in a new top of book data block being sent. This is due to the transitory nature of the event. However, last best price is used to capture this information and convey it to the market.

FIX Syntax for Last Best Price (Bid or Ask) - Market Data Incremental Refresh (tag 35-MsgType =X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	0 or 1	0 = bid and 1 = ask. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
276	QuoteCondition	C	C = exchange best. Space-delimited list of conditions describing a quote.

Tag 277-TradeCondition is not sent for this data block.

Market Data - Trade

The trade data block is sent when a trade occurs to provide volume and trade statistics. The following sections detail various trade types and their respective data blocks that demonstrate the use of Tag 277-TradeCondition.

Tag 277-TradeCondition is present and set to 1 when the reported trade price is an assigned price by CME Globex.

Tag 277-TradeCondition is not present when the reported trade price is not assigned by CME Globex.

When Tag 277-TradeCondition is not present or Tag 277-TradeCondition=E (opening trade), then the data represents the last trade.

Tag 269-MDEntryType=2 (trade) is not used to update the view of the book.

Tag 5797-AggressorSide is sent in Market Data Incremental Refresh (tag 35-MsgType=X) messages for trades where one side (buy or sell) is the aggressing customer-submitted order. Trades without aggressors occur at Market Open, after a Pre-Open or after a Pause. Trades without aggressors also occur when the triggering order is a CME Globex-generated implied bid/offer. For a spread trade, tag 5797-AggressorSide is sent on the Market Data Incremental Refresh (tag 35-MsgType=X) messages for both the spread and the legs.

> ⚠ An implied order cannot be the aggressor in a trade. Tag 5797-AggressorSide is not sent in an implied spread or implied leg trade message.

A trade data block may also be flagged as either the beginning or the end of a CME Globex event.

See also:

- Trade - Two Outright Orders Trading
- Trade - Two Spread Orders Trading - Not Implied
- Trade - Spread Order Trading Against Two Outright Orders, Implied
- Trade - Butterfly Order Trading Against a Calendar Order and Two Outright Orders, Implied

Trade - Two Outright Orders Trading

The following example illustrates the trade data block that is sent when a buy and sell match on GEZ2.

FIX Syntax for GEZ2 Outright Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.

451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.
5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.
5799	MatchEventIndicator	1 or 2	1 = Beginning and 2 = End. Indicates the beginning or the end of a CME Globex event. If there is no value present, then the message is not at the beginning or the end of a CME Globex event. Currently, only 1 is a valid value. 2 will become a valid value in a future release.

Tag 277-TradeCondition is not sent for this data block.

Trade - Two Spread Orders Trading - Not Implied

Several trade data blocks are sent for a spread. A trade may be executed in an off-market manner, and in this case the leg trade prices are CME-calculated and may not reflect the current market prices.

The following example illustrates the data blocks that are sent for a match event involving one order on ZCM8-ZCU8 against one order on ZCM8-ZCU8.

The resulting trade data blocks that will occur are:

- One trade data block for ZCM8ZCU8 (tag 277-TradeCondition is not sent)
- One trade data block for ZCM8 (tag 277-TradeCondition = 1)
- One trade data block for ZCU8 (tag 277-TradeCondition = 1)

FIX Syntax for ZCM8-ZCU8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	1360.00	Price of the Market Data Entry.

48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.
5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.
5799	MatchEventIndicator	1 or 2	1 = Beginning and 2 = End. Indicates the beginning or the end of a CME Globex event. If there is no value present, then the message is not at the beginning or the end of a CME Globex event. ⓘ Currently, only 1 is a valid value. 2 will become a valid value in a future release.

ⓘ Tag 277-TradeCondition is not sent for this data block.

FIX Syntax ZCM8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	1360.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.
277	TradeCondition	1	Space-delimited list of conditions describing a trade. 1 = Price calculated by CME Globex. Tag 277- TradeCondition is not sent for a last best price data block.
5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.

FIX Syntax ZCU8 Trade - Market Data Incremental Refresh (tag 35-MsgType =X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	1360.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.

277	TradeCondition	1	Space-delimited list of conditions describing a trade. 1 = Price calculated by CME Globex. Tag 277 - TradeCondition is not sent for a last best price data block.
5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.

Trade - Spread Order Trading Against Two Outright Orders, Implied

The following example illustrates the data blocks that are sent for a match event involving one order on GEM8-GEU8 against one order on GEM8 and one order on GEU8.

The resulting trade data blocks that will occur are:

- One trade data block for GEM8-GEU8 (tag 277-TradeCondition is not sent)
- One trade data block for GEM8 (tag 277-TradeCondition is not sent)
- One trade data block for GEU8 (tag 277-TradeCondition is not sent)

FIX Syntax for GEM8-GEU8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.

5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.
5799	MatchEventIndicator	1 or 2	1 = Beginning and 2 = End. Indicates the beginning or the end of a CME Globex event. If there is no value present, then the message is not at the beginning or the end of a CME Globex event. ⓘ Currently, only 1 is a valid value. 2 will become a valid value in a future release.

ⓘ Tag 277-TradeCondition is not sent for this data block.

FIX Syntax GEM8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.

ⓘ Tag 277-TradeCondition is not sent for this data block.

FIX Syntax GEU8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.

Tag 277-TradeCondition is not sent for this data block.

Trade - Butterfly Order Trading Against a Calendar Order and Two Outright Orders, Implied

The following example illustrates the data blocks that are sent for a match event involving one order on GE:BF M8-U8-Z8 against one order on GEM8-GEU8, one order on GEU8, and one order on GEZ8.

The resulting trade data blocks that will occur are:

- One trade data block for GE:BF M8-U8-Z8 (tag 277-TradeCondition is not sent)
- One trade data block for GEM8-GEU8 (tag 277-TradeCondition is not sent)
- One trade data block for GEM8 (tag 277-TradeCondition = 1)
- One trade data block for GEU8 (tag 277-TradeCondition = 1)
- One trade data block for GEU8 (tag 277-TradeCondition is not sent)
- One trade data block for GEZ8 (tag 277-TradeCondition is not sent)

FIX Syntax GE:BF M8-U8-Z8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.

269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.
5797	AggressorSide	1 or 2	1 = Buy and 2 = Sell. Indicates if the aggressor is on the buy or sell side of the order on the book that triggers a trade.
5799	MatchEventIndicator	1 or 2	1 = Beginning and 2 = End. Indicates the beginning or the end of a CME Globex event. If there is no value present, then the message is not at the beginning or the end of a CME Globex event.

Currently, only 1 is a valid value. 2 will become a valid value in a future release

FIX Syntax GEM8-GEU8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.

48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.

Tag 277-TradeCondition is not sent for this data block.

FIX Syntax GEM8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.

277	TradeCondition	1	Space-delimited list of conditions describing a trade. 1 = Price calculated by CME Globex. Tag 277 - TradeCondition is not sent for a last best price data block.

FIX Syntax for GEU8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.
277	TradeCondition	1	Space-delimited list of conditions describing a trade. 1 = Price calculated by CME Globex. Tag 277 - TradeCondition is not sent for a last best price data block.

FIX Syntax GEU8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.

Tag 277-TradeCondition is not sent for this data block.

FIX Syntax GEZ8 Trade - Market Data Incremental Refresh (tag 35-MsgType = X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0 or 2	0 = new and 2 = delete. Type of Market Data update action.
269	MDEntryType	2	2 = trade. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
271	MDEntrySize	5	Quantity or volume represented by the Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.

22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
1020	TradeVolume		Total traded volume since the beginning of the session.
274	TickDirection		Direction of the tick. If there is no value present, then there is no change.
451	NetChgPrevDay		Net change from previous day's closing price versus last traded price.

Tag 277-TradeCondition is not sent for this data block.

Market Data - Session High and Low Trade Price

The High Trade Price data block is sent when a trade occurs which results in the highest trade price for the current session. Likewise, the Low Trade Price data block is sent when a trade occurs which results in the lowest trade price for the current session. High and low trade prices are helpful in tracking market trends. They also provide historical information for the current session regarding market behavior.

CME Group will provide high and low trade prices only as a response to the event that produces them.

FIX Syntax for Session High Trade Price - Market Data Incremental Refresh (tag 35-MsgType=X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	7	7 = session high. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

FIX Syntax for Session Low Trade Price - Market Data Incremental Refresh (tag 35-MsgType=X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	8	8 = session low. Type of Market Data entry.

83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

Market Data - High Bid and Low Ask

The high bid and low ask prices are used to indicate the highest bid and lowest ask prices of the trading session and are useful in tracking market behavior.

The high bid and low ask can be different from the high trade and low trade price if an arriving bid or ask order price is better than the specified limit order.

> Aggressing orders that execute immediately do not rest on the book and are not transmitted as a high bid or low offer.

FIX Syntax for Best High Bid - Market Data Incremental Refresh (tag 35-MsgType=X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	N	N = session high bid. Type of Market Data entry.
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

FIX Syntax for Best Low Ask - Market Data Incremental Refresh (tag 35-MsgType=X)

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	O	O = session low ask. Type of Market Data entry.

83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

Market Data - Open Interest

Open Interest is sent using Market Data Incremental Refresh (tag 35-MsgType=X) message data blocks which contain the total number of contracts per instrument that are not yet offset or fulfilled for the previous trading day. They are sent before the start of the trading session. Tag 75-TradeDate will contain the date of the previous trading day. Tag 279-MDUpdateAction will be set to 0 = New and tag 269-MDEntryType will be set to C = Open Interest. Tag 271-MDEntrySize will contain the quantity or volume of the Open Interest.

Open Interest is updated throughout the trading week. Client systems can leverage the Security Definition (tag 35-MsgType=d) message to recover the last published open interest for an instrument. With this release, on Sunday start, the Security Definition (tag 35-MsgType=d) Message is published initially to the Instrument Definition feeds and the Market Data Incremental Refresh feeds without tag 5792-OpenInterestQty. Once open interest is sent in the Market Data Incremental Refresh (tag 35-MsgType=X) Message, the value of open interest from tag 271-MDEntrySize is placed on the Security Definition (tag 35-MsgType=d) message in tag 5792-OpenInterestQty and maintained on the Instrument Definition feed only. As open interest updates are disseminated on the Market Data Incremental Refresh feed throughout the week, the value of tag 5792-OpenInterestQty is updated.

If open interest is 0, tag 5792-OpenInterestQty=0.

Open Interest Data Block Sample Message

The following sample Market Data Incremental Refresh (tag 35-MsgType=X) message contains the Open Interest data block:
1128=8|35=X|49=CME|34=6957050|52=20080124064553918|75=20080124|268=1|279=0|22=8|48=64288|83=127|269=C|271=2000|273=064553000|

Tag	FIX Name	Format	Sample Values	Description
FIX Header 1128=8|9=118|35=X|49=CME|34=6957050|52=20080124163153918|75=20080124|268=1|				
279	MDUpdateAction	Char (1)	0	Type of Market Data update action. 0 = New
22	SecurityIDSource	String (1)	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48- SecurityID is specified.

48	SecurityID	Int (12)	64288	Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
83	RptSeq	Int (1)	127	Sequence number per Instrument update.
269	MDEntryType	Char (1)	C	Type of Market Data Entry. C = Open Interest
271	MDEntrySize	Qty(12)	2000	Quantity or volume represented by the Market Data Entry.
273	MDEntryTime	UTCTimeOnly (12)	64553000	Time of Market Data Entry. The Market Data entry time is 06:45:53.000 UTC time

Market Data - Cleared Volume

Cleared Volume data blocks contain the number of contracts that have been through the clearing process for an active instrument for the previous trading day. Tag 75-TradeDate will contain the current trade date. Tag 279-MDUpdateAction will be set to 0 = New and tag 269-MDEntryType will be set to B = Trade Volume. Tag 271-MDEntrySize will contain the Cleared Volume.

Cleared Volume is updated throughout the trading week. Client systems can leverage the Security Definition (tag 35-MsgType=d) Message to recover the last published cleared volume for an instrument. On Sunday start, the Security Definition (tag 35-MsgType=d) Message is published initially to the Instrument Definition feeds and the Market Data Incremental Refresh feeds without tag 5791-ClearedVolume. Once Cleared Volume is sent in the Market Data Incremental Refresh (tag 35-MsgType=X) message, the value of Cleared Volume from tag 271-MDEntrySize is placed on the Security Definition (tag 35-MsgType=d) Message in tag 5791-ClearedVolume and maintained on the Instrument Definition feed. As cleared volume updates are disseminated on the Market Data Incremental Refresh feed throughout the week, the value of tag 5791-ClearedVolume is updated.

If cleared volume is 0, tag 5791-ClearedVolume=0.

Cleared Volume Data Block Sample Message

The following sample Market Data Incremental Refresh (tag 35-MsgType=X) message contains the Cleared Volume data block.

1128=8|35=X|49=CME|34=6957050|52=20080124163153918|75=20080124|268=1|279=0|22=8|48=6428 8|83=127|269=B|271=2000|273=163153000|

Tag	FIX Name	Format	Sample Values	Description
FIX Header 1128=8|35=X|49=CME|34=6957050|52=20080124163153918|75=20080124|268=1|				
279	MDUpdateAction	Char (1)	0	Type of Market Data update action. 0 = New

22	SecurityIDSource	String (1)	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.
48	SecurityID	Int (12)	64288	Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
83	RptSeq	Int (1)	127	Sequence number per Instrument update.
269	MDEntryType	Char (1)	B	Type of Market Data Entry. B = Trade Volume
271	MDEntrySize	Qty(12)	2000	Quantity or volume represented by the Market Data Entry.

Market Data - Fixing Price

The Fixing Price data block contains the Fixing Price, a volume-weighted average price for the nearby futures contract and the time the Fixing Price was calculated (tag 5790-FixingBracket). Tag 279-MDUpdateAction will be set to 0 = New and tag 269-MDEntryType will be set to W = Fixing Price.

Fixing Price Data Block Sample Message

The following Market Data Incremental Refresh (tag 35-MsgType=X) sample message contains the Fixing Price data block.

1128=8|35=X|49=CME|34=6957050|52=20080124134553918|75=20080124|268=1|279=0|22=8
269=W|270=2000|273=134553000|5790=14:00

Tag	FIX Name	Format	Sample Values	Description
FIX Header 1128=8|35=X|49=CME|34=6957050|52=20080124134553918|75=20080124|268=1|				
279	MDUpdateAction	Char (1)	0	Type of Market Data update action. 0 = New
22	SecurityIDSource	String (1)	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48- SecurityID is specified.
48	SecurityID	Int (12)	64288	Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
83	RptSeq	Int (1)	127	Sequence number per Instrument update.
269	MDEntryType	Char (1)	W	Type of Market Data Entry. W = Fixing Price
270	MDEntryPx	Char (1)	2000	Price of Market Data Entry.

273	MDEntryTime	UTCTimeOnly (12)	134553000	Time of Market Data Entry. The Market Data entry time is13:45:53.000 UTC time
5790	FixingBracket	Char (5)	14:00	Time that the Fixing Price was calculated.

Market Data - Settlement Price

Settlement price is updated throughout the trading week for an instrument and disseminated by means of the Market Data Incremental Refresh (tag 35-MsgType=X) message on the Market Data Incremental Refresh feed and the Security Definition (tag 35-MsgType=d) message for the instrument on the Instrument Definition feed.

Incremental Refresh (35=X) Settlement Price

CME Globex sends the following types of settlement prices in the Market Data Incremental Refresh (tag 35-MsgType=X) message:

- Preliminary Settlement
  - Actual Preliminary Settlement
  - Rounded Preliminary Settlement
- Final Settlement
  - Final Theoretical Settlement

tag	FIX Name	Format	Valid Values	Description
269	MDEntryType	char(1)	6=settlement price	identifies price as a settlement price
286	OpenCloseSettleFlag	int(3)	• 5=theoretical price value • 100 = Actual Preliminary settlement price for instruments not subject to settlement price rounding OR Rounded Preliminary for instruments subject to settlement rounding • 101 = Actual Preliminary settlement price	Indicates settlement price type This tag is not not sent for final settlement.

Security Definition (35=d) Settlement Price

When an update occurs to a settlement price, that price is disseminated on the Security Definition (tag 35-MsgType=d) message for the instrument on the Instrument Definition feed. For the Security Definition (35=d) message, as new settlement prices are published, the date (tag 5796-TradingReferenceDate) and price (tag 1150-TradingReferencePrice) are updated. Tag 731-SettlPriceType in the Security Definition (tag 35-MsgType=d) message indicates the settlement price type.

On Sunday start, the Security Definition (tag 35-MsgType=d) message is published initially to the Instrument Definition feeds and the Market Data Incremental Refresh feeds without tag 5796-TradingReferenceDate. The settlement price in tag 1150-TradingReferencePrice will reflect the last published settlement price for the settlement date in tag 5796-TradingReferenceDate.

Tag	FIX Name	Format	Valid Values	Description
731	SettlPriceType	String (3)	1=Final 2 = Theoretical 100 = Actual Preliminary settlement price for instruments not subject to settlement price rounding OR Rounded Preliminary for instruments subject to settlement rounding 101 = Actual Preliminary settlement price	Indicates whether settlement price is preliminary or final.
1150	TradingReferencePrice	price(20)		Settlement price
5796	TradingReferenceDate	LocalMktDate(8)		Indicates the date the last update to the settlement price (tag 1150-TradingReferencePrice) was updated in this message in YYYYMMDD format. Absence of this field indicates current day (expressed in local time at place of trade).

For instruments subject to settlement price rounding and having an on-tick Actual Preliminary, the settlement price is the same for tag 731-SettlPriceType=100 and tag 731-SettlePriceType=101.

Settlement Date for Sunday Start

Currently, on Sunday start, the Security Definition (tag 35-MsgType=d) message is published initially to the Market Data Incremental Refresh channels and the Instrument Definition loop channels without tag 5796-TradingReference-Date.

Tag 5796-TradingReference-Date will be added to the security definition messages to reflect the settlement date for the settlement price in tag 1150-TradingReferencePrice published initially to the Market Data Incremental Refresh channel on Sunday start.

The Initial Security Definition messages on Sunday start contain the following tags and values:

Security Definition (tag 35-MsgType=d) messages are published initially to the Incremental Refresh channels with:

- Tag 731-SettlPriceType

- Tag 1150-TradingReferencePrice
- Tag 5796-TradingReferenceDate

Market Data - Final Settlement

Final Settlement

The final daily settlement prices are the official daily settlements published by CME Clearing and disseminated after the start of the next trading day on CME Globex, Monday through Thursday between 5:30 p.m. and 9:30 p.m. CT.

The final settlement data blocks contain the following tags and values:

- Market Data Incremental Refresh (tag 35-MsgType=**X**) messages are sent on the Incremental Refresh channels with:
  - Tag 269-MDEntryType=**6**
  - Tag 270-MDEntryPX
  - Tag 64-SettlDate

Note: tag 286-OpenCloseSettleFlag will not be present.

The security definition messages are updated with the following tags and values:

- Security Definition (tag 35-MsgType=**d**) messages are updated on the Instrument Definition loop channels with:
  - Tag 731-SettlPriceType=**1**
  - Tag 1150-TradingReferencePrice
  - Tag 5796-TradingReferenceDate

Because final settlement prices for some instruments are not available until after 5:30 p.m. CT, not all final settlement prices are published in the Market Data Incremental Refresh (tag 35=X) on Fridays. Client systems should leverage either the Market Data Incremental Refresh feeds or Instrument Definition feeds upon Sunday startup to determine the most recent final settlement price.

Final Theoretical Settlement

The final theoretical settlement data blocks contain the following tags and values:

- Market Data Incremental Refresh (tag 35-MsgType=**X**) is sent on the on the Incremental Refresh channels with:
  - Tag 269-MDEntryType=**6**
  - Tag 286-OpenCloseSettleFlag=**5**
  - Tag 270-MDEntryPX
  - Tag 64-SettlDate

The security definition messages are updated with the following tags and values:

- Security Definition (tag 35-MsgType=**d**) is updated on the Instrument Definition loop channels with:
  - Tag 731-SettlPriceType=**2**
  - Tag 1150-TradingReferencePrice
  - Tag 5796-TradingReferenceDate

For more information on settlements, refer to the Settlement Prices.

Market Data - Preliminary Settlement

The preliminary settlement price can be sent multiple times per session per product and is subject to change. A preliminary settlement price remains subject to change pending transmission of the Final settlement price.

Market Data Messaging

The following provides a summary of message tags used to transmit Preliminary settlement

market data in the Market Data - Incremental Refresh and Market Data - Security Definition messages.

Market Data Incremental Refresh (tag 35-MsgType=X)

- Tag 269-MDEntryType=**6** (Settlement price)
- Tag 286-OpenCloseSettleFlag=

  - 100 - Actual Preliminary settlement price for instrument not subject to settlement price rounding (i.e. standard or 'Big' contract) or Rounded Preliminary for instrument subject to settlement rounding (i.e. 'mini' or 'micro' contract settlement price rounded to correspond to 'Big' tick increment)

  - 101 - Actual Preliminary settlement price for instrument subject to settlement price rounding (i.e. 'mini' or 'micro' contract) but not rounded.

- Tag 270-MDEntryPX
- Tag 64-SettlDate

Security Definition (tag 35-MsgType=d) messages are updated on the Instrument Definition loop channels with:

- Tag 731-SettlPriceType=**100** (Actual Preliminary settlement price for instruments not subject to settlement price rounding OR Rounded Preliminary for instruments subject to settlement rounding) or **101** (Actual Preliminary settlement price)
- Tag 1150-TradingReferencePrice
- Tag 5796-TradingReferenceDate

For more information on settlements, refer to Settlement Prices.

Preliminary Settlement for Mini-Sized Products

The e-mini, e-micro, and miNY products settle to the price of their corresponding full-sized product. However, if the full-sized product has a smaller minimum tick increment than the corresponding e-mini, e-micro, or miNY product, the preliminary settlement price of the mini-sized contract may be off-tick. Therefore, all products that settle to another product's settlement price are evaluated for settlement price rounding.

Example

In the following example, the E-mini S&P 500 Future settles to the S&P 500 Future settlement price:

Product	Minimum Tick Increment
S&P 500 Future (tag 1151-SecurityGroup=SP)	0.10
E-mini S&P 500 Future (tag 1151-SecurityGroup=ES)	0.25

1. The S&P 500 Future settles at 1213.60.
2. The E-mini S&P 500 Future settlement price of 1213.60 is sent and marked as an Actual preliminary settlement price (tag 286-OpenCloseSettleFlag=101).
3. The settlement price 1213.60 is evaluated for rounding and must be rounded to 1213.50 to align with the E-mini S&P 500 Future minimum tick increment.
4. The E-mini S&P 500 Future settlement price of 1213.50 is sent and marked as a Rounded preliminary settlement price (tag 286-OpenCloseSettleFlag=100).

When the Actual preliminary settlement price is on-tick for the e-mini, e-micro, and miNY products, CME Globex will continue to send a Rounded preliminary settlement price with the identical settlement price. For example:

1. The S&P 500 Future settles at 1213.50.
2. The E-mini S&P 500 Future settlement price of 1213.50 is sent and marked as an Actual preliminary settlement price (tag 286-OpenCloseSettleFlag=101).
3. The settlement price 1213.50 is evaluated for rounding and is not required to align with the E-mini S&P 500 Future minimum tick increment.
4. The E-mini S&P 500 Future settlement price of 1213.50 is sent and marked as a Rounded preliminary settlement price (tag 286-OpenCloseSettleFlag=100).

All other ("non-rounded") products are settled to their own trade prices and are always on-tick. For these products, only the Actual Preliminary is sent. For non-rounded products, that is, those that do not have a settlement price rounding evaluation, tag 286-OpenCloseSettleFlag=100 indicates an Actual Preliminary settlement price. Tag 286-OpenCloseSettleFlag=101 is not sent for these products since their settlement prices are always on-tick.
Refer to the following link for more information on settlement prices: http://www.cmegroup.com/market-data/settlements/.

The following tables show the tags included in the Market Data Incremental Refresh (tag 35-MsgType=X) message for each settlement price type.

E-mini, e-micro, and miNY Products: Rounded Preliminary Price

Sent for a mini-sized product preliminary settlement price rounded to the tick increment of the larger-sized contract.

Tag	FIX Name	Format	Sample Value	Description
75	TradeDate	LocalMktDate (8)	20100615	Indicates current session date in YYYYMMDD format.
64	SettleDate	LocalMktDate (8)	20100614	Settlement Date to which this price applies in YYYYMMDD format.
286	OpenCloseSettleFlag	Int (3)	100	Flag to indicate the type of entry 100 = Rounded Preliminary
269	MDEntryType	Char (1)	6	Type of Market Data entry. 6 = Settlement Price

E-mini, e-micro, and miNY Products: Actual Preliminary Price

Sent for a mini-sized product settlement price which did not require rounding (i.e. Actual).

Tag	FIX Name	Format	Sample Value	Description
75	TradeDate	LocalMktDate (8)	20100615	Indicates current session date in YYYYMMDD format.
64	SettleDate	LocalMktDate (8)	20100614	Settlement Date to which this price applies in YYYYMMDD format.
286	OpenCloseSettleFlag	Int (3)	101	Flag to indicate the type of entry 101 = Actual Preliminary (non-rounded)

269	MDEntryType	Char (1)	6	Type of Market Data entry. 6 = Settlement Price

Non-Rounded Products: Preliminary Settlement

Sent for the standard or large-sized contract settlement price.

Tag	FIX Name	Format	Sample Value	Description
75	TradeDate	LocalMktDate (8)	20100615	Indicates current session date in YYYYMMDD format.
64	SettleDate	LocalMktDate (8)	20100614	Settlement Date to which this price applies in YYYYMMDD format.
286	OpenCloseSetleFlag	Int (3)	100	Flag to indicate the type of entry 100=Actual Preliminary
269	MDEntryType	Int (1)	6	Type of Market Data entry. 6 = Settlement Price

Market Data - Trading Statistics Reset Based on Market State Value

Statistics are sent on the Incremental Feed with the Market Data Incremental Refresh (tag 35-MsgType=X) message as they occur throughout a trading session. Certain statistics are also made available on the Market Recovery Feed with the Market Data Snapshot Full Refresh (tag 35-MsgType=W) message. Statistics are removed at the end of a trading session from the Market Recovery Feed on the Market Data Snapshot Full Refresh (tag 35-MsgType=W) message.

Tag 1174-SecurityTradingEvent=4 in the Security Status (tag 35-MsgType=f) Message indicates the end of a trading session, and triggers the process to remove statistics from the Market Recovery Feed on the Market Data Snapshot Full Refresh (tag 35-MsgType=W) message.

The following process takes place:

1. A Security Status (tag 35-MsgType=f) message is sent, containing the following tags:
   - Tag 326-SecurityTradingStatus can be any value.
   - Tag 1174-SecurityTradingEvent set to 4 = End of trading
2. CME Globex removes the following statistics from the Market Recovery Feed on the Market Data Snapshot Full Refresh (tag 35-MsgType=W) message:
   - Session High Bid (tag 269-MDEntryType=N)
   - Session Low Offer (tag 269-MDEntryType=O)
   - Opening Price (tag 269-MDEntryType=4)
   - Trading Session High Price (tag 269-MDEntryType=7)
   - Trading Session Low Price (tag 269-MDEntryType=8)
   - Trade Volume (tag 1020-TradeVolume) for the Last Trade data block (tag 269-MDEntryType=2)

   > The Last Trade (tag 269-MDEntryType=2) statistic is not removed from the Market Recovery Feed. It is not removed until a new Last Trade is generated.
3. Client systems can choose to reset the statistics.

Statistics will not be sent in the Market Data Snapshot Full Refresh (tag 35-MsgType=W) message on the Market Recovery feed until activity on the Incremental feed creates new statistics.

Tag 75-TradeDate indicates the session date of the trade referenced in this message.

Tag 64-SettleDate indicates the settlement date for the price in tag 1150-TradingReferencePrice.

Market Data - Opening Traded Price

FIX Syntax for Opening Traded Price - Market Data - Incremental Refresh

Tag Number	Tag Name	Value	Description
279	MDUpdateAction	0	0 = new. Type of Market Data update action.
269	MDEntryType	2	Trade
83	RptSeq		Sequence number per Instrument update.
273	MDEntryTime		Time of Market Data Entry.
270	MDEntryPx	9550.00	Price of the Market Data Entry.
48	SecurityID		Unique instrument ID as qualified by the exchange per tag 22-SecurityIDSource.
277	TradeCondition	E	Sent for opening trade.
22	SecurityIDSource	8	Identifies source of tag 48-SecurityID value. This value is always 8 for CME and is required if tag 48-SecurityID is specified.

Market Data - Recovery

CME Group offers several options for recovering missed messages or synchronizing client systems to the latest state: Market Recovery, Instrument Replay, and TCP Replay. Instrument level sequencing and natural refresh can be utilized to supplement the recovery process.

This topic contains an overview of recovery methods and supplemental methods including:

- Market Data - Recovery Feeds
- Market Data - Recovery - Using the Incremental Market Data Feed to Determine State
- Market Data - Order Book Reset
- Market Recovery and Instrument Definition Rates

Notes:

- CME Group strongly recommends that client systems process both the A and B Incremental UDP feeds. UDP Feed A and UDP Feed B provide the first level of protection against missed market data messages.
- CME Group recommends Market Recovery in conjunction with Natural Refresh as a primary recovery option.
- All customers must certify for Market Recovery functionality.

Market Data - Recovery Feeds

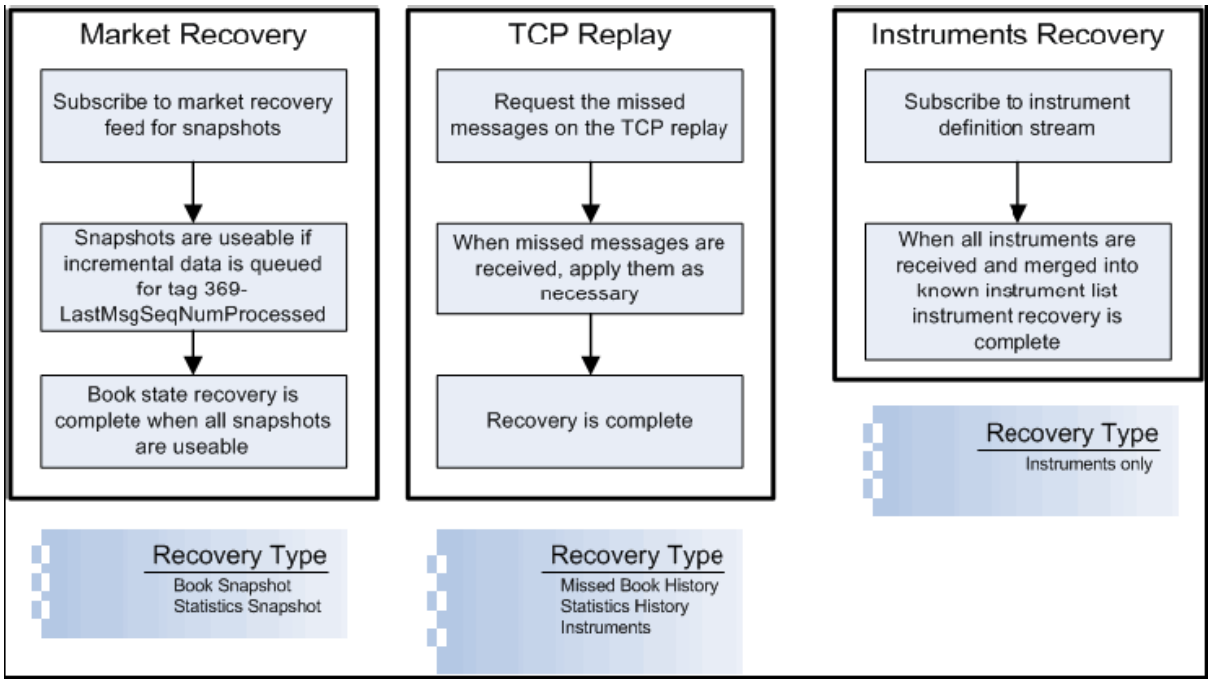
UDP Feed A and UDP Feed B are used to disseminate CME Group incremental market data using bandwidth efficient FAST encrypted FIX messages. All FIX message types are sent through UDP Feed A and UDP Feed B applicable Market Data Groups (book update, statistics, quotes, instrument definitions, trades, instrument / instrument group, CME Globex status). Please note that a single FIX/FAST message can contain multiple updates for multiple instruments. In some instances, a market data message may be missed from UDP Feed A but will be received from UDP Feed B.

In some instances, messages may be missed on both feeds, requiring a recovery process to take place. Message loss is detected using the decoded FIX message sequence numbers (tag 34-MsgSeqNum), which are also found in the Preamble sequence number. The message sequence number is an incrementing number, therefore, if a gap is detected between messages in the tag 34-MsgSeqNum value, or the Preamble sequence number, this indicates a message has been missed. In addition, tag 83-RptSeq can be used to detect a gap between the messages at the instrument level. It should be assumed at this point that all books maintained in the client system may no longer have the correct, latest state maintained by CME Group. Client systems must resynchronize all books to the latest state maintained by CME Group, and determine whether any new instrument definitions were published. During this synchronization process, all books are initially assumed to be in an incorrect state and are recovered during the synchronization process.

All book updates in the latest Market Data Incremental Refresh message (tag 35-MsgType=X) messages must be processed before the order book can be considered valid.

The following options are offered by CME Group to recover missed messages or synchronize client systems to the latest state:

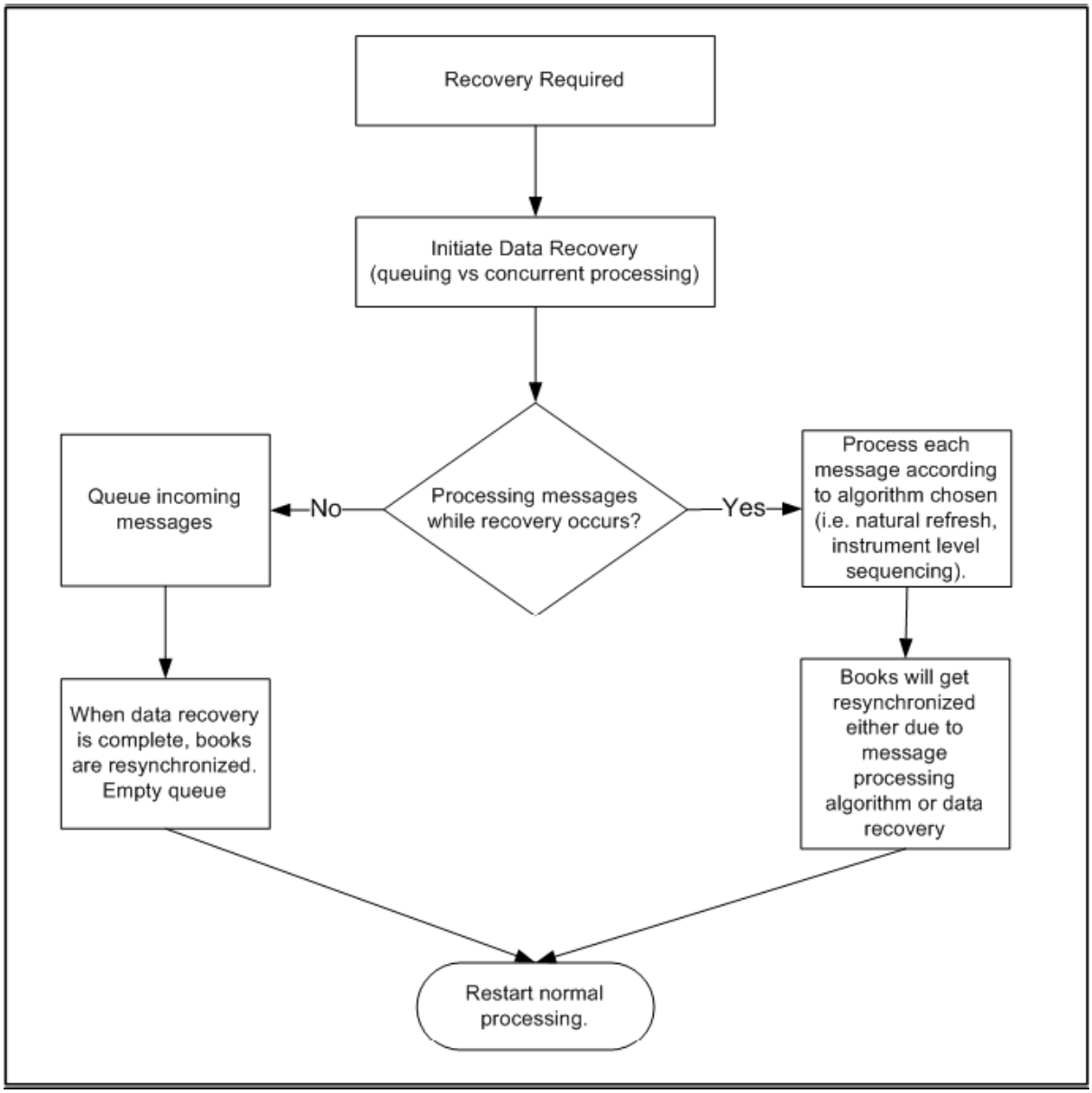
- Market Recovery
- Instrument Replay
- TCP Replay


Market Recovery
Subscribe to market recovery feed for snapshots
Snapshots are useable if incremental data is queued for tag 369-LastMsgSeqNumProcessed
Book state recovery is complete when all snapshots are useable
Recovery Type
Book Snapshot
Statistics Snapshot
TCP Replay
Request the missed messages on the TCP replay
When missed messages are received, apply them as necessary
Recovery is complete
Recovery Type
Missed Book History
Statistics History
Instruments
Instruments Recovery
Subscribe to instrument definition stream
When all instruments are received and merged into known instrument list instrument recovery is complete
Recovery Type
Instruments only

Market Data - Recovering Data - Process

This section describes CME Group recommended recovery processes. Based on the recovery and supplemental methods, client systems can choose to implement those methods as applicable. In general, two paths can be followed: processing while recovering and queuing while recovering.

> The recovering data process applies to affected channels only. Unaffected channels can continue normal processing.


Recovery Required
Initiate Data Recovery
(queuing vs concurrent processing)
Processing messages while recovery occurs?
No
Yes
Queue incoming messages
Process each message according to algorithm chosen (i.e. natural refresh, instrument level sequencing).
When data recovery is complete, books are resynchronized. Empty queue
Books will get resynchronized either due to message processing algorithm or data recovery
Restart normal processing.

The following sections describe process recovery options in various failure scenarios:

- Large Scale Outage Using Market Recovery – Queuing
- Large Scale Outage Using Market Recovery - Concurrent Processing

> The Market Recovery feed should be used for large-scale data recovery (i.e. major outage or late joiners). TCP replay is not a performance-based recovery option and should only be

used for small-scale data recovery.

Large Scale Outage Using Market Recovery – Queuing

This section describes the process to follow for a large scale outage in which the client system is out of sync. This process uses Market Recovery - queuing of the Incremental Market Data feed and Market Recovery feed until the client system is synchronized to the latest state advertised by CME Group. In order to avoid an excessive number of queued messages, process snapshots and apply the applicable incremental feed as the snapshots arrive.

If the Market Recovery method is used, client systems also need to subscribe to the Instrument Definition feed to determine whether any new instruments were defined. Client systems will not recover any missed statistics on the Market Recovery feed.

1. Identify channel(s) in which the client system is out of sync.
2. Listen to and queue the Incremental Market Data feed for the affected channel(s).
3. Listen to the Market Recovery feed for the affected channel(s).
4. Verify that all snapshots have been received for a given Market Recovery feed.

   Tag 911-TotNumReports in a Market Data Snapshot Full Refresh (tag 35-MsgType=W) message contains the total number of messages that have been sent. Listen to the Market Recovery feed until the indicated number of Market Data Snapshot Full Refresh (tag 35-MsgType=W) messages has been received.
5. Apply all incremental data in sequence, where tag 34-MsgSeqNum (or the Preamble sequence number) is greater than the lowest value for tag 369-LastMsgSeqNumProcessed

   OR

   where tag 83-RptSeq from the Market Data Incremental Refresh (tag 35-MsgType=X) message is greater than the lowest value for tag 83-RptSeq on the Market Recovery feed.
6. Using queued data, restart normal processing.

Large Scale Outage Using Market Recovery - Concurrent Processing

This section describes at the process to follow for a large scale outage using Market Recovery while continuing to process the Incremental Market Data feed and obtaining snapshots from the Market Recovery feed. Once books are recovered, they can resume normal processing even if other books are still being recovered.

If the Market Recovery method is used, client systems also need to subscribe to the Instrument Definition feed to determine whether any new instruments were defined.

Client systems will not recover any missed statistics on the Market Recovery feed.

1. Identify channel(s) in which the client system is out of sync.
2. Listen to the Incremental Market Data feed for the affected channel(s) and optionally attempt a natural refresh.
3. Listen to the Market Recovery feed for the affected channel(s).

4. For each book, compare tag 369-LastMsgSeqNumProcessed on the Market Recovery feed to tag 34-MsgSeqNum (or the Preamble sequence number) on the Incremental Market Data feed and verify that the value for tag 34-MsgSeqNum is not lower

   OR

   compare tag 83-RptSeq on the Market Recovery feed to tag 83-RptSeq on the Incremental Market Data feed and verify that the value for tag 83-RptSeq on the Incremental Market Data feed is not lower.

5. Restart normal processing.

Market Data - Market Recovery

This recovery method should be used for large-scale data recovery (i.e. major outage or late joiners) to synchronize client systems to the latest state maintained by CME Group. Client systems can use the Market Recovery feed on each channel to determine the state of each book in every channel. Each Market Recovery feed constantly loops and sends the Market Data Snapshot Full Refresh (tag 35-MsgType=W) message.

⚠ Instrument updates can occur during processing of of a Snapshot loop. Once a loop is processed, it should be compared against the Instrument Recovery cache.

The Market Recovery feed is valid as of the corresponding instrument level sequence number on the Incremental Market Data feed. This instrument level sequence number (tag 83-RptSeq) is found on each Market Data Incremental Refresh Message (tag 35-MsgType=X) on the Incremental Market Data feed. It is also found on every Market Data Snapshot Full Refresh (tag 35-MsgType=W) message on the Market Recovery feed. As a result, it can be determined which book updates were missed for a specific instrument.

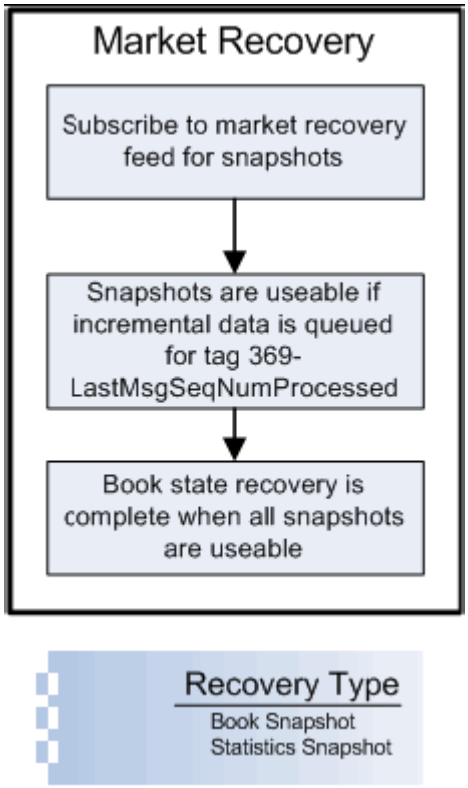
The Market Recovery feed is also valid as the corresponding sequence number on the Incremental Market Data feed. The sequence number (tag 369-LastMsgSeqNumProcessed) on the Market Recovery feed corresponds to the sequence number [(tag 34-MsgSeqNum) or the Preamble sequence number] on the Incremental Market Data feed. Tag 369-LastMsgSeqNumProcessed is contained in each Market Data Snapshot Full Refresh (tag 35=W) message and tag 34-MsgSeqNum is contained in each Market Data Incremental Refresh message. The sequence number can be used to determine which book updates were missed for a specific instrument.

If the Market Recovery method is used, client systems also need to subscribe to the Instrument Definition feed to determine whether any new instruments were defined. Client systems will recover the most recent statistics on the Market Recovery feed. Any intermediary statistics will not be recovered.

Market Recovery in conjunction with Natural Refresh provides exceptional performance and reliable recovery of market data.

ⓘ Client systems should queue real-time data until all snapshot data is retrieved. The queued data should then be applied as necessary.

ⓘ CME Group requires that the Market Recovery feeds be used for recovery purposes only. Once client systems have retrieved recovery data, client systems should stop listening to the Market Recovery feeds.


Market Recovery
Subscribe to market recovery feed for snapshots
Snapshots are useable if incremental data is queued for tag 369-LastMsgSeqNumProcessed
Book state recovery is complete when all snapshots are useable
Recovery Type
Book Snapshot
Statistics Snapshot

Tag 83-RptSeq provides an instrument level sequence number and can be leveraged instead of tag 369-LastMsgSeqNumProcessed.

Market Recovery Scenarios

The following examples depict the relation between Incremental Feed and Snapshot Loop sequence numbers and how to recover in the given scenario.

Scenario 1: tag 369-LastMsgSeqNumProcessed is consistent in the Snapshot Loop, Customer Recovers from beginning of the loop

Incremental Feed	Snapshot Loop			
tag 34-MsgSeqNum	Packet Sequence Number	tag 369-LastMsgSeq NumProcessed	tag 911-TotNum Reports	
10100 drop	7	10098	10	
10101 drop	8	10098	10	
10102 drop	9	10098	10	
10103	10	10098	10	
10104 (start processing from 10104)	1	10103	10	Process full loop
10105	2	10103	10	
	3	10103	10	

	4	10103	10	
	5	10103	10	
	. . .			
	10	10103	10	

Scenario 2: tag 369-LastMsgSeqNumProcessed is consistent in the Snapshot Loop, customer recovers from middle of the loop

Incremental Feed	Snapshot Loop			
tag 34-MsgSeqNum	Packet Sequence Number	tag 369-LastMsgSeqNumProcessed	tag 911-TotNumReports	
10100	3	10103	10	Start processing mid-loop
10101	4	10103	10	
10102	5	10103	10	
10103	. . .			
10104	10	10103	10	Because tag 369 value changed, process full loop once more OR obtain difference from queue by checking tag 83-RptSeq value.
10105	1	10105	12	
10106	2	10105	12	
10107	3	10105	12	
10108	. . .			
	12	10105	12	

Scenario 3: tag 369-LastMsgSeqNumProcessed is inconsistent in the Snapshot Loop, customer recovers from the start of the loop

In this scenario, notice that new instruments are not included in the Snapshot Loop; therefore the client system must read tag 83-RptSeq from the queue—instrument by instrument—and process the messages that were missed.

Incremental Feed				Snapshot Loop			
tag 34-MsgSeqNum	tag 35-MsgType	tag 48-SecurityID	tag 83-RptSeq	Packet Sequence Number	tag 369-LastMsgSeqNumProcessed	tag 911-TotNumReports	tag 48-Security
10100 drop				7	10098	10	
10101 drop				8	10098	10	
10102 drop				9	10098	10	
10103	X	111 222	12390 8908	10	10098	10	

10104 process	f			next loop			
10105	X	123 555	89891 1	1	10103	10	111
10106	X	333 111	3321 12391	2	10103	10	123
10107 process	Process blocks in bold only			3	10104	10	222
10108 process				4	10104	10	223
				5	10105	10	333
				. . .			
				10	10106	10	555

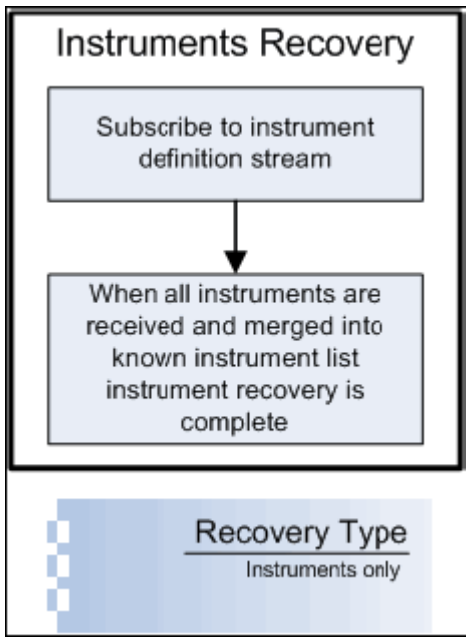
Scenario 4: tag 369-LastMsgSeqNumProcessed is inconsistent in the Snapshot Loop, client system recovers from the middle of the loop

Incremental Feed				Snapshot Loop			
tag 34-MsgSeqNum	tag 35-MsgType	tag 48-SecurityID	tag 83-RptSeq	Packet Sequence Number	tag 369-LastMsgSeq NumProcessed	tag 911-TotNumReports	tag 48-Security
10100 drop				. . .			
10101 drop				3	10104	11	222
10102 drop				4	10104	11	223
10103 drop	X	111 222	12390 8908	5	10105	11	333
10104 process	f			. . .			
10105	X	123 555	89891 1	12	10106	11	555
10106	X	333 111	3321 12391	1	10107	13	111
10107	X	777 555	345 2	2	10107	13	123
10108	X	555 123 888	3 89892 4	3	10108	13	222

10109 process	d	Process blocks in bold only	4	10108	13	223
10110 process	f		5	10108	13	333
10111 process	X		. . .			
			12	10108	13	555
			13	10108	13	777

Market Data - Instrument Recovery

Client systems can use the Instrument Replay feed on each channel to determine if any new instruments have been defined through the Security Definition (tag 35-MsgType=d) message. The Instrument Replay feed constantly replays the current week's Security Definition (tag-MsgType=d) messages, and will reflect any additions, modifications, or deletes to available instruments. Client systems should begin the recovery process when tag 34-MsgSeqNum = 1. The value in tag 911-TotNumReports indicates the total number of messages to process for a complete set of Security Definitions. CME Group does not guarantee the order of the messages being sent.


Instruments Recovery
Subscribe to instrument definition stream
When all instruments are received and merged into known instrument list instrument recovery is complete
Recovery Type
Instruments only

Market Data - Message Recovery over TCP

TCP replay should be used for small-scale data recovery (refer to Implementation Considerations for additional information). Client systems can recover specific messages that were missed using the sequence number and the TCP historical replay component. This method recovers all missed messages.CME Group logs IP addresses, ports, and passwords of the originator and the Channel ID and range of the requested sequence number.

> TCP Replay is not a performance-based recovery option and should only be used if other options are unavailable.

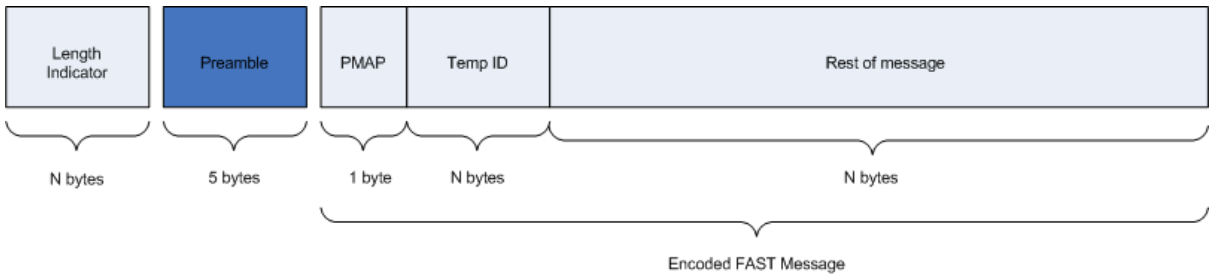
The TCP historical replay component allows you to request a replay of a set of messages already published on the UDP Incremental Market Data Channel. The request specifies messages to be replayed. The request is submitted with the FIX Market Data Request message (35=V). In response, CME Group sends the requested messages and a Logout (35=5) message. A Logout (35=5) message is also sent in case of rejection of the initial request, tag 58-Text will list the reason for the logout.

This type of request is sent through a new TCP connection established by the customer. When making the request, the channel and sequence number are specified. The responses are sent by CME Group through this same connection and the connection is then closed by CME Group once the replay is complete. A maximum of 2,000 messages can be requested at once.

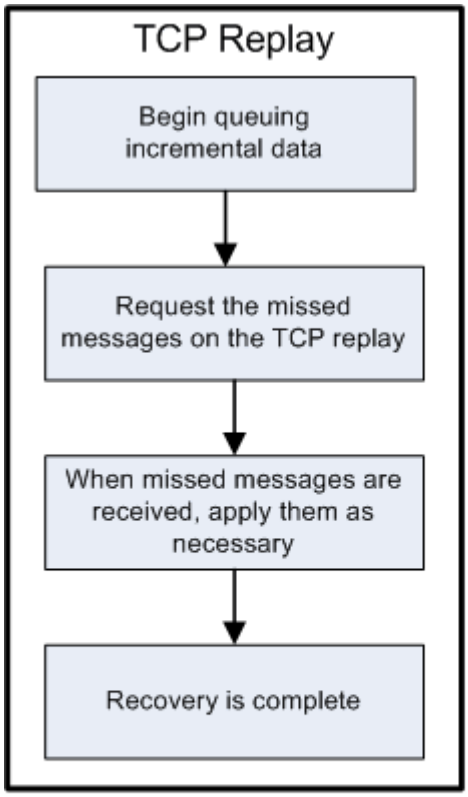
The TCP Replay feed contains 3 different message formats:

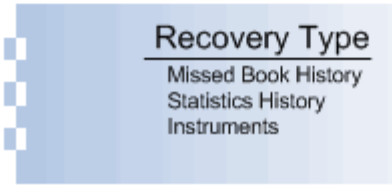
- **Raw FIX message** - Logon (tag 35-MsgType=A) - Customer to CME Group, Logout (tag 35-MsgType=5) - Customer to CME Group, and FIX Market Data Request message (35=V).
- **Length Indicator + Generated Preamble + FAST message** - Logon (tag 35-MsgType=A) - CME Group to Customer, Logout (tag 35-MsgType=5) - CME Group to Customer, Heartbeat (tag 35-MsgType=0).
- **Length Indicator + Original Preamble + FAST message** - All requested messages for recovery.

The Length Indicator is a stop-bit encoded integer. Replayed FIX/FAST messages will contain their original Preambles.



Client systems should queue real-time data until all missed data is recovered. The recovered data should then be applied prior to queued data.


TCP Replay
Begin queuing incremental data
Request the missed messages on the TCP replay
When missed messages are received, apply them as necessary
Recovery is complete



TCP Replay - Implementation Considerations

The following are limitations to consider prior to implementing TCP replay functionality:

- TCP Replay is not a performance-based recovery option and should only be used if other options are unavailable.
- TCP requests and responses are via TCP.
- Resend request messages and other customer-generated messages (i.e. heartbeat) are in FIX format, CME Globex-generated response messages are FAST encoded.
- There is a 24 hour time limit on the messages available via the TCP replay functionality.

Only one Market Data Request can be processed during the current established session. If multiple Market Data Requests are sent when the session is established, only the first Market Data Request is processed and subsequent Market Data Requests are ignored.

To submit multiple market data requests, for each request, the client system must log in, submit the request, receive the messages, log out, and repeat until all requests are processed.

- The maximum number of messages that can be requested in one resend request message is 2000.

TCP Replay - Process

1. Customer establishes a TCP connection.

   Refer to the configuration file for TCP IP and port information. For additional information on the configuration file, refer to Market Data - Services.

2. Customer sends Logon (tag 35-MsgType=A) message to CME Group.

   > Customer Username and Password are verified. If the Username and Password are incorrect, a Logout will be sent.

3. CME Group sends Logon (tag 35-MsgType=A) message to the customer.

   > CME Group will send a Logout message if a Market Data Request (tag35-MsgType=V) message is not received in 5 seconds.

4. Customer sends Market Data Request (tag35-MsgType=V) message to CME Group.

   > Client systems must indicate the channel ID (tag 1180-ApplFeedID) and sequence numbers (tag 1182-ApplBeginSeqNo and tag 1183-ApplEndSeqNo) in the Market Data Request (tag 35-MsgType=V) message.

5. CME Group sends Heartbeat (tag35-MsgType=0) messages to customer every 2 seconds until the first recovery message is sent.

6. CME Group sends FIX/FAST recovery messages that were requested by the customer in the Market Data Request (tag35-MsgType=V) message.

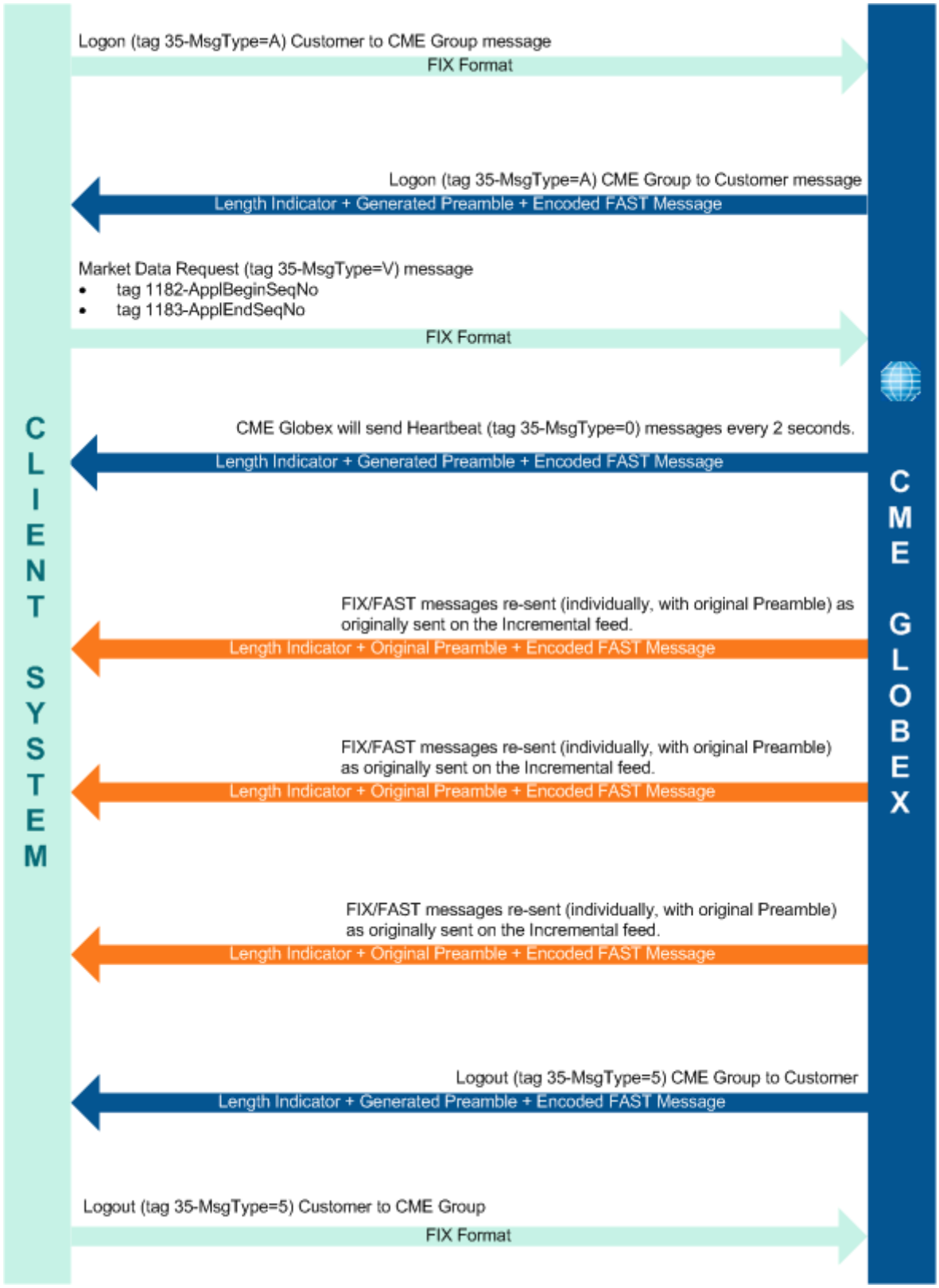
7. CME Group sends a Logout (tag35-MsgType=5) message to the customer.

   > CME Group will close the TCP connection if the customer does not send a Logout (tag 35-MsgType=5) message within 5 seconds from the time CME Group sends a Logout (tag35-MsgType=5) message.

8. Customer sends a Logout (tag 35-MsgType=5) message to CME Group.

9. CME Group closes the TCP connection.

   > The username and password for the TCP Replay component for the New Release and Certification testing environments are both "CME" (do not include quotes). The username and password for the production environment are provided after certification is completed. For more information, contact your Global Account Manager.


Logon (tag 35-MsgType=A) Customer to CME Group message
FIX Format
Logon (tag 35-MsgType=A) CME Group to Customer message
Length Indicator + Generated Preamble + Encoded FAST Message
Market Data Request (tag 35-MsgType=V) message
tag 1182-ApplBeginSeqNo
tag 1183-ApplEndSeqNo
FIX Format
CME Globex will send Heartbeat (tag 35-MsgType=0) messages every 2 seconds.
Length Indicator + Generated Preamble + Encoded FAST Message
FIX/FAST messages re-sent (individually, with original Preamble) as originally sent on the Incremental feed.
Length Indicator + Original Preamble + Encoded FAST Message
FIX/FAST messages re-sent (individually, with original Preamble) as originally sent on the Incremental feed.
Length Indicator + Original Preamble + Encoded FAST Message
FIX/FAST messages re-sent (individually, with original Preamble) as originally sent on the Incremental feed.
Length Indicator + Original Preamble + Encoded FAST Message
Logout (tag 35-MsgType=5) CME Group to Customer
Length Indicator + Generated Preamble + Encoded FAST Message
Logout (tag 35-MsgType=5) Customer to CME Group
FIX Format
CLIENT SYSTEM
CME GLOBEX

Market Data - Recovery - Using the Incremental Market Data

Feed to Determine State

In addition to the recovery methods that are offered by CME Group, there are additional mechanisms that client systems can utilize to enhance the recovery process.

Instrument Level Sequencing
Natural Refresh

Market Data - Instrument Level Sequencing

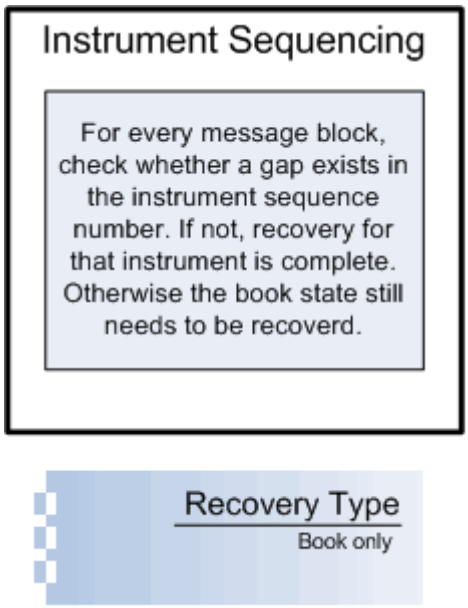
Market Data Incremental Refresh (tag 35-MsgType=X) messages contain instrument sequence numbers (tag 83-RptSeq), in addition to message sequence numbers (tag 34-MsgSeqNum). Every repeating group instance of a market data entry contains an incrementing sequence number (tag 83-RptSeq) that is associated with the instrument for which data is present in the block.

> Customers must support Market Recovery functionality.

Client systems can keep track of the instrument sequence number (tag 83-RptSeq) for every instrument by inspecting incoming data and determining whether there is a gap in the instrument sequence number (tag 83-RptSeq).

- If there is a gap in the instrument sequence number (tag 83-RptSeq), it indicates that data was missed for the instrument when message loss occurred.
- If there is no gap, the data can be used immediately, and it also indicates that the book for this instrument still has a correct, current state.

It is likely that if only a small number of messages have been missed, there will be data in subsequent messages which are not affected by the missing data. If there are 10 instruments in a channel, for example, and the missed messages contain data for 2 of these instruments, any subsequent messages containing data about the other instruments are still valid.


Instrument Sequencing
For every message block, check whether a gap exists in the instrument sequence number. If not, recovery for that instrument is complete. Otherwise the book state still needs to be recoverd.
Recovery Type
Book only

Market Data - Natural Refresh

The client system must track the state of the book at all times with the FIX Market Data

Incremental Refresh message (tag 35-MsgType=X) book update data blocks. It is possible, though not guaranteed, that a set of these book update data blocks can be used to construct the current, correct state of a book without prior book state knowledge. While client systems wait for the recovery of missing data, they may opt to also use a natural refresh algorithm to recover book state prior to recovering the lost data. Natural refresh can also be used to re-instate the top-of-book. Natural refresh should be used for multiple depth book updates only in conjunction with market recovery. Prior to beginning a natural refresh, the entire book should be emptied. Natural refresh assumes no prior knowledge of book state.

The following example illustrates how natural refresh occurs. Note that in this example, using instructions 4 and 5, the top two levels of the offer book are known to be good. This portion of the book can then be displayed or used.

> Natural Refresh is not guaranteed and should not be considered a definitive substitute for recovering lost data. Natural Refresh should only be used in conjunction with Market Recovery.

Example:
In this example, the book is first cleared.

Bid		Ask	
Quantity	Price	Price	Quantity

Then, the following data blocks are processed for the given instrument with a 5-deep book.

1. Insert at Bid book level 3, price 40, quantity 10:

tag 279-MDUpdateAction = 0 (new)
tag 1023-MDPriceLevel = 3
tag 269-MDEntryType = 0 (Bid)
tag 271-MDEntrySize = 10
tag 270-MDEntryPx = 40

Bid			Ask
Quantity	Price	Price	Quantity
10	40		

2. Insert at Ask book level 3, price 70, quantity 10:

tag 279-MDUpdateAction = 0 (new)
tag 1023-MDPriceLevel = 3
tag 269-MDEntryType = 1 (Offer)
tag 271-MDEntrySize = 10

tag 270-MDEntryPx = 70

Bid			Ask
Quantity	Price	Price	Quantity
10	40	70	10

3. Insert at Bid book level 2, price 45, quantity 10:

tag 279-MDUpdateAction = 0 (new)
tag 1023-MDPriceLevel = 2
tag 269-MDEntryType = 0 (Bid)
tag 271-MDEntrySize = 10
tag 270-MDEntryPx = 45

Bid			Ask
Quantity	Price	Price	Quantity
10	45		
		70	10
10	40		

4. Insert at Ask book level 2, price 60, quantity 10:

tag 279-MDUpdateAction = 0 (new)

tag 1023-MDPriceLevel = 2

tag 269-MDEntryType = 1 (Offer)

tag 271-MDEntrySize = 10

tag 270-MDEntryPx = 60

Bid			Ask
Quantity	Price	Price	Quantity
10	45	60	10
10	40	70	10

5. Insert at Ask book level 1, price 55, quantity 10:

tag 279-MDUpdateAction = 0 (new)
tag 1023-MDPriceLevel = 1
tag 269-MDEntryType = 1 (Offer)
tag 271-MDEntrySize = 10
tag 270-MDEntryPx = 55

Bid			Ask
Quantity	Price	Price	Quantity
		55	10
10	45		
		60	10
10	40		
		70	10

6. Update Bid book level 1, price 50, quantity 10:

tag 279-MDUpdateAction = 1 (change)
tag 1023-MDPriceLevel = 1
tag 269-MDEntryType = 0 (Bid)
tag 271-MDEntrySize = 10
tag 270-MDEntryPx = 50

Bid			Ask
Quantity	Price	Price	Quantity
10	50	55	10
10	45		
		60	10
10	40		
		70	10

Market Data - Order Book Reset

Order Book Reset provides a process for synchronizing order books in the unlikely event of a CME Group dual component failure. In this scenario, order books on the channel may be "corrupted".

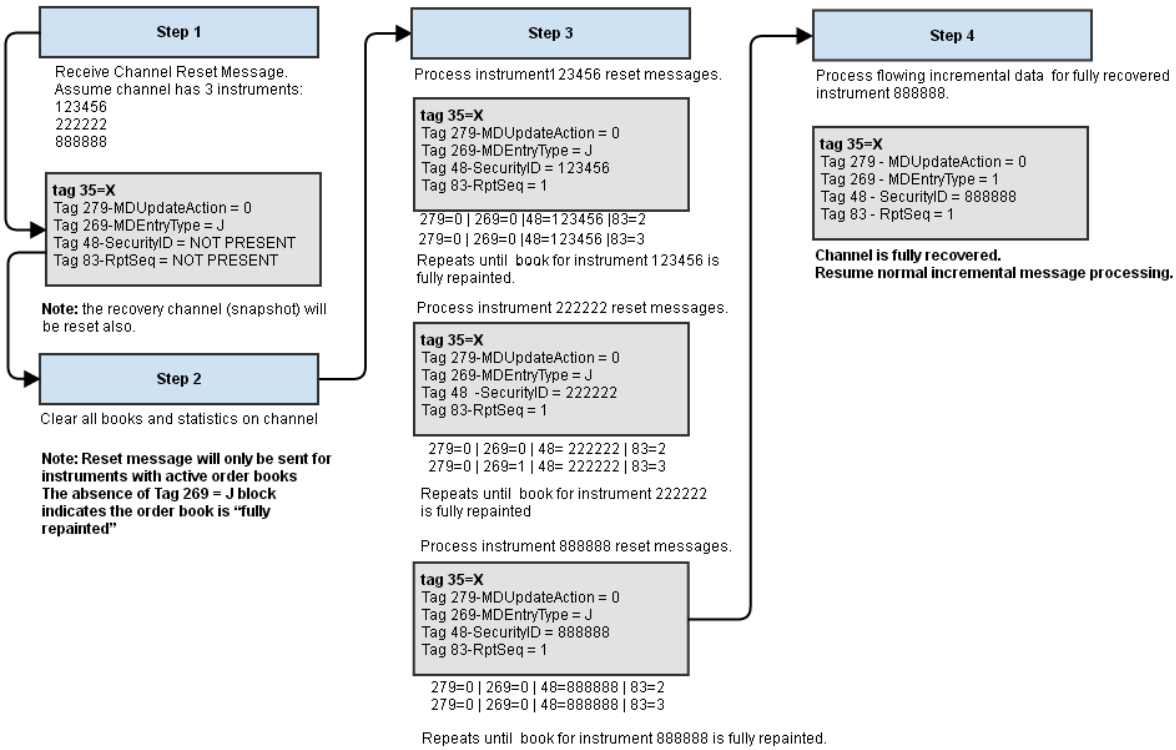
A tag 269-MDEntryType=J value in the Market Data Incremental Refresh (tag 35-MsgType=X) message will identify an Order Book Reset occurrence.

Market Data Incremental Refresh (tag 35-MsgType=X) Message

Tag	FIX Name	Format	Valid Values	Description
269	MDEntryType	Char (1)	0 = Bid 1 = Offer 2 = Trade 4 = Opening Price 6 = Settlement Price 7 = Trading Session High Price 8 = Trading Session Low Price E = Simulated Sell F = Simulated Buy J = Order Book Reset N = Session High Bid O = Session Low Offer	Type of Market Data entry.

Order Book Reset Process

The steps to detect this condition and recover are shown below:


Step 1
Receive Channel Reset Message.
Assume channel has 3 instruments:
123456
222222
888888
tag 35=X
Tag 279-MDUpdateAction = 0
Tag 269-MDEntryType = J
Tag 48-SecurityID = NOT PRESENT
Tag 83-RptSeq = NOT PRESENT
Note: the recovery channel (snapshot) will be reset also.
Step 2
Clear all books and statistics on channel
Note: Reset message will only be sent for instruments with active order books
The absence of Tag 269 = J block indicates the order book is "fully repainted"
Step 3
Process instrument123456 reset messages.
tag 35=X
Tag 279-MDUpdateAction = 0
Tag 269-MDEntryType = J
Tag 48-SecurityID = 123456
Tag 83-RptSeq = 1
279=0 | 269=0 |48=123456 |83=2
279=0 | 269=0 |48=123456 |83=3
Repeats until book for instrument 123456 is fully repainted.
Process instrument 222222 reset messages.
tag 35=X
Tag 279-MDUpdateAction = 0
Tag 269-MDEntryType = J
Tag 48 -SecurityID = 222222
Tag 83-RptSeq = 1
279=0 | 269=0 | 48= 222222 | 83=2
279=0 | 269=1 | 48= 222222 | 83=3
Repeats until book for instrument 222222 is fully repainted
Process instrument 888888 reset messages.
tag 35=X
Tag 279-MDUpdateAction = 0
Tag 269-MDEntryType = J
Tag 48-SecurityID = 888888
Tag 83-RptSeq = 1
279=0 | 269=0 | 48=888888 | 83=2
279=0 | 269=0 | 48=888888 | 83=3
Repeats until book for instrument 888888 is fully repainted.
Step 4
Process flowing incremental data for fully recovered instrument 888888.
tag 35=X
Tag 279 - MDUpdateAction = 0
Tag 269 - MDEntryType = 1
Tag 48 - SecurityID = 888888
Tag 83 - RptSeq = 1
Channel is fully recovered.
Resume normal incremental message processing.

1. The Market Data Incremental Refresh (tag 35-MsgType=X) message, Order Book Reset data block is sent down the Incremental feed with tag 269-MDEntryType=J to indicate that there has been a dual component failure and the order books on the channel are corrupted. For the FAST message that contains the initial Order Book Reset data block, the Preamble will be denoted as administrative.
2. The client system must empty all order books for the impacted channel.

When the Order Book Reset data block is received, no order entry actions should be taken based on the data found in the order book. The order book and all statistics for the instrument should be emptied on the client system.

3. The Market Data Snapshot Full Refresh (tag 35-MsgType=W) message on the Market Recovery feed will be deleted for impacted instruments.
4. Market Data Incremental Refresh (tag 35-MsgType=X) messages will be sent to populate all price levels in the order books.

   *The first data block will contain tag 269-MDEntryType=J and tag 48-SecurityID, followed by data blocks for each level of the order book as identified in tag 48-SecurityID.
   *Client systems should leverage tag 269-MDEntryType=J in combination with tag 48-SecurityID to determine that a book for the instrument identified in tag 48-SecurityID has been recovered.

Once a book has been recovered for an instrument (identified by tag 48-SecurityID), CME Globex will disseminate incremental real-time market data for that instrument, and client systems can process data for that instrument. At this point, other instruments on the channel may still be going through the recovery process.

Tag 83-RptSeq will reset to 1 for each instrument repeating group.

Market Recovery and Instrument Definition Rates

The following table contains the Market Recovery MPS Settings for FIX/FAST Production.

Channel	Definition	MPS Setting
7	CME Globex Equity Futures	100
8	CME Globex Equity Options	480
9	CME Globex Interest Rate Futures	400
10	CME Globex Interest Rate Options	400
11	CME Globex FX Futures	100
12	CME Globex FX Options	400
13	CME Globex Commodity Futures	300
14	CME Globex Commodity Options	100
21	KRX Globex Futures	50
28	DME Globex Futures	100
30	NYMEX Globex Crude & Crude Refined Futures	2000
31	NYMEX Globex Nat Gas & other Non-Crude Energy Futures	150
32	NYMEX Globex Metals, Softs, & Alternative Market Futures	150
35	NYMEX Globex Crude & Crude Refined Options	400
36	NYMEX Globex Nat Gas & other Non-Crude Energy Options	300

37	NYMEX Globex Metals, Softs, & Alternative Market Options	100
111	CBOT Globex Commodity Futures	150
112	CBOT Globex Commodity Options	400
113	CBOT Globex Equity Index Futures	60
114	CBOT Globex Equity Index Options	400
115	CBOT Globex Interest Rate Futures	100
116	CBOT Globex Interest Rate Options	400

The following table contains the Instrument Definition MPS Settings for FIX/FAST Production.

Channel	Definition	MPS Setting
7	CME Globex Equity Futures	20
8	CME Globex Equity Options	500
9	CME Globex Interest Rate Futures	100
10	CME Globex Interest Rate Options	500
11	CME Globex FX Futures	100
12	CME Globex FX Options	600
13	CME Globex Commodity Futures	250
14	CME Globex Commodity Options	500
15	CME Globex Equity Futures - excludes E-mini S&P 500	25
16	CME Globex Equity Options - excludes E-mini S&P 500	600
18	NYMEX Globex Emissions Futures	25
19	NYMEX Globex Emissions Options	200
21	KRX Globex Futures	20
22	BMD Globex Futures	50
23	BMD Globex Options	50
24	BM&F Globex Futures	50
25	BM&F Globex Options	100
26	MexDer Globex Futures	50
27	MexDer Globex Options	25
28	DME Globex Futures	200

29	DME Globex Options	300
30	NYMEX Globex Crude & Crude Refined Futures	800
31	NYMEX Globex Nat Gas & other Non-Crude Energy Futures	1,000
32	NYMEX Globex Metals, Softs, & Alternative Market Futures	100
33	COMEX Globex Futures	150
35	NYMEX Globex Crude & Crude Refined Options	2,000
36	NYMEX Globex Nat Gas & other Non-Crude Energy Options	800
37	NYMEX Globex Metals, Softs, & Alternative Market Options	25
38	COMEX Globex Options	600
60	CME Globex FX Futures II	70
61	CME Globex FX Options II	600
70	CME Europe Globex Futures	100
111	CBOT Globex Commodity Futures	100
112	CBOT Globex Commodity Options	400
113	CBOT Globex Equity Index Futures	20
114	CBOT Globex Equity Index Options	100
115	CBOT Globex Interest Rate Futures	50
116	CBOT Globex Interest Rate Options	500
117	CBOT Globex Commodity Futures II	100
118	MGEX Globex Futures	100
119	MGEX Globex Options	100

CME Globex Legacy Market Data to MDP 3.0 Market Data Conversion

CME Group completed the launch of Market Data Platform (MDP) 3.0 on Sunday, December 7, 2014. Market Data Platform (MDP) 3.0 is a low latency and direct data access solution designed to improve customers' ability to process and act on real-time price information.

- Certification Requirements
- Feed of Record Information
- MDP3 Overview - FIX/FAST to MDP 3.0
- FIX/FAST to MDP3 Channel Mapping
- Market Data Message Impacts
- Market Data Incremental Refresh (tag 35-MsgType=X) Message
- Market Data Incremental Refresh Trade Summary Message
  - Trade Summary Opt-In
- Market Data Security Status (tag 35-MsgType=f) Message
- Quote Request (tag 35-MsgType=R) Message
- Market Data Snapshot Full Refresh (tag 35-MsgType=W) Message
- Market Data Security Definition (tag 35-MsgType=d) Message
- Unchanged Messages
- Recovery
  - Channel Reset
- Order Routing Partner Exchanges
- Streamlined FIX/FAST

Certification Requirements

All customers must certify for, and migrate to, MDP 3.0 by October 2015.

Feed of Record Information

Currently, in case of market data failures, markets may be halted for either FIX/FAST or MDP 3.0 issues. CME Group will review each event individually to determine the best response.

In September 2015, MDP 3.0 will become the sole "Feed of Record" and markets will be halted for MDP 3.0 issues only. If there are issues with FIX/FAST only, after this date, the markets will not be halted.

Important Dates	
MDP 3.0 Sole Feed of Record – All Channels	September 2015
All FIX/FAST Channels Decommissioned	October 2015

TAS Pre-Validation

Currently, order entry will be permitted in TAS (Trade at Settlement) markets as soon as the Pre-Open state message is published, regardless of whether it is received first via the FIX/FAST or MDP 3.0 feeds. As of the MDP 3.0 Feed of Record date, only the publishing of the MDP 3.0 Pre-Open state message will be valid for the start of the TAS market and TAS orders can be entered. The CME Group Market Regulation rules for TAS Transactions can be found here.

MDP3 Overview - FIX/FAST to MDP 3.0

The following topics summarize the features of MDP3 and impacts for client systems transitioning from the FIX/FAST format to MDP3.

The SBE format is a fixed position, fixed field length protocol supporting direct access to data, therefore avoiding the need for management of multiple variable-length elements which must be sequentially processed. This makes it easier for client systems to scan the message for key fields to determine whether or not to process the message.

FIX/FAST to MDP3 Channel Mapping

Channels from the FIX/FAST market data dissemination map to the MDP3 implementation as follows.

MDP 3.0 Channel	FIX/FAST Channel	Channel Label	Designated Contract Market (DCM)
310	7	CME Globex Equity Futures	CME
311	8	CME Globex Equity Options	
312	9	CME Globex Interest Rate Futures	
313	10	CME Globex Interest Rate Options	
314	11	CME Globex FX Futures	
315	12	CME Globex FX Options	
316	13	CME Globex Commodity Futures	
317	14	CME Globex Commodity Options	
318	15	CME Globex Equity Futures - excludes E-mini S&P 500	
319	16	CME Globex Equity Options - excludes E-mini S&P 500	
320	60	CME Globex FX Futures II	
321	61	CME Globex FX Options II	
340	111	CBOT Globex Commodity Futures	CBOT
341	112	CBOT Globex Commodity Options	
342	113	CBOT Globex Equity Index Futures	
343	114	CBOT Globex Equity Index Options	
344	115	CBOT Globex Interest Rate Futures	
345	116	CBOT Globex Interest Rate Options	
346	117	CBOT Globex Commodity Futures II	
360	33	COMEX Globex Futures	COMEX

361	38	COMEX Globex Options	
380	18	NYMEX Globex Emissions Futures	NYMEX
381	19	NYMEX Globex Emissions Options	
382	30	NYMEX Globex Crude & Crude Refined Futures	
383	35	NYMEX Globex Crude & Crude Refined Options	
384	32	NYMEX Globex Metals, Softs, & Alternative Market Futures	
385	37	NYMEX Globex Metals, Softs, & Alternative Market Options	
386	31	NYMEX Globex Nat Gas & other Non-Crude Energy Futures	
387	36	NYMEX Globex Nat Gas & other Non-Crude Energy Options	
410	70	CME Europe Globex Futures	CME Europe
430	22	BMD Globex Futures	Hosted Partner Exchanges
431	23	BMD Globex Options	
440	28	DME Globex Futures	
441	29	DME Globex Options	
450	21	KRX Globex Futures	
460	118	MGEX Globex Futures	
461	119	MGEX Globex Options	

Market Data Message Impacts

MDP 3.0 uses the same message types currently used for MDP FIX/FAST, but the FIX message layouts have changed.

The following market data messages are disseminated by CME Group:

- Heartbeat (tag 35-MsgType=0)
- Logon (tag 35-MsgType =A)
- Logout (tag 35-MsgType =5)
- Market Data Incremental Refresh (tag 35-MsgType =X)
- Security Status (tag 35-MsgType =f)
- Request for Quote (tag 35-MsgType =R)
- Security Definition (tag 35-MsgType =d)
- Snapshot Full Refresh (tag 35-MsgType =W)
- Replay Request (tag 35-MsgType =V)

The following changes were made to market data distribution:

- More than one message may be included in a packet (a packet is a formatted unit of data carried on a network). The maximum size of a packet is 1420 bytes.
- Multiple messages will only be included in a packet if they are available at the same time and will fit into the packet.
- A single FIX message will not be split over multiple packets
- Updates for a given event may be split into multiple FIX messages and possibly several packets

Market Data Incremental Refresh (tag 35-MsgType=X) Message

The following changes were made to the Market Data Incremental Refresh (35=X) message:

- The Start of Event Timestamp (tag 60-TransactTime) is sent in nanoseconds with guaranteed microsecond precision.
- The End of Event Indicator (tag 5799-MatchEventIndicator) – specifies that the event has ended and all data related to the event has been sent. It also indicates when each type of message block update is complete. The tag value consists of eight bits, representing the following:

  Bit 0 (least significant bit): Last trade summary message for a given event

  Bit 1: Last electronic volume message for a given event

  Bit 2: Last real quote message for a given event

  Bit 3: Last statistic message for a given event

  Bit 4: Last implied quote message for a given event

  Bit 5: Message resent during recovery

  Bit 6: Reserved for future use

  Bit 7 (most significant bit): Last message for a given event

> The End of Event indicator is always sent. In certain scenarios, it can be sent in a Market Data Incremental Refresh (35=X) message with no data blocks (268=0).

- New Entry Types were added for implied bids/offers. The following tag 269-MDEntryType values were added:

  - E = Implied Bid

  - F = Implied Offer

- Additional Market Data Types (Tag 269-MDEntryType) were added to support updates to Trade Volume and real-time changes to Price Bands and Limits:

  - e = Electronic Volume

  - g = Threshold Limits

- Electronic Volume is generated whenever trading volume is updated.

  - Electronic Volume is included in a message only for the instruments that have traded as part of the event.

  - Trade Summary is sent when at least one real (not implied) order has participated in the trade.

  - Events may include implied orders and leg trades where actual trade summary messages will not be included, but Electronic Volume will be updated to reflect the quantity traded.

- Two values for tag 279-MDUpdateAction were added to save bandwidth:

  - Delete Thru (tag 279=3) - indicates that all positions on one side of the book should be deleted. Tag 1023-MDPriceLevel will always equal 1 in the Delete Thru block.

- Delete From (tag 279=4) - indicates that all positions from the position stated in tag 1023-MDPriceLevel to position 1 should be deleted. This situation occurs when several levels of the book are executed due to a large aggressing order.

- The following statistic data blocks include the trading date:

  - Trade Volume

  - Open Interest

  - Settlement

  - Fixing Price

- Tag 731-SettlPriceType contains a bitmap field of Boolean type indicators representing the settlement price type.

  Bit 0 (least significant bit):

  - 1=Final
  - 0=Preliminary

  - Bit 1:

    - 1=Actual
    - 0=Theoretical or Undefined

      Note: Typically, there are two rounds of preliminary settlement prices disseminated. The early round of preliminary settlement prices will have Bit 1 set to 0 (Undefined).

  - Bit 2:

    - 1= Rounded
    - 0= Non-Rounded or Undefined

  - Bit 3:

    - 1= Intraday
    - 0 = Undefined

  - Bit 4: Reserved for future use, set to 0

  - Bit 5: Reserved for future use, set to 0

  - Bit 6: Reserved for future use, set to 0

  - Bit 7 (most significant bit):

    - 0=not NULL
    - 1=entire set is a NULL

Tag 286-OpenCloseSettlFlag is no longer sent for settlement prices.

The following data was removed from the FIX/FAST Market Data Incremental Refresh (35=X) message format:

- Tag 75-TradeDate - only the time is included on each message to save space. The Trade Date is included on every Security Status (35=f) message update.
- Tag 273-MDEntryTime - Currently, tag 273-MDEntryTime is sent in each Market Data Incremental Refresh (35=X) message data block. Beginning with this release, the Start of Event timestamp (tag 60-TransactTime) is sent once for each message. This provides a more accurate timestamp for the message blocks.
- The following market data is no longer sent:

- Better Bid/Offer (CLAST)
- Simulated Buy/Sell
- Calculated Prices for Trades (Leg or Implied Orders Only)
- Net Change
- Tick Direction
- Trade Condition

- Tag 731-SettlPriceType is used for settlement prices; tag 286-OpenCloseSettlFlag is no longer sent for settlement prices.

Market Data Incremental Refresh Trade Summary Message

The Market Data Incremental Refresh Message includes a Trade Summary message which replaces the current Trade message. The Trade Summary indicates the quantity and optional inclusion of the anonymous, CME Globex-assigned order identifier.

Trade Summary Opt-In

A new FIX tag in Order Entry iLink messages allows customers to opt-out or display tag 37-OrderID in the Market Data Incremental Trade Summary message (Tag 35=X). This provides customers with the option of disclosing their order ID to the marketplace for any order they submit.

Tag	Name	FIX Type	Enumeration	Req	Description
1091	PreTradeAnonymity	Boolean	Y N	N	Defines whether the tag 37-OrderID is included in the Trade Summary message in MDP 3.0 If tag is not sent or sent with “Y”, tag 37 will not appear in the Trade Summary message. If tag is sent with “N”, tag 37 will appear in the Trade Summary message

Tag 1091-PreTradeAnonymity can be included in one of the following messages. This field is optional, therefore if it is not included in the Incoming message, it will not be included in the Outgoing message.

Incoming

- New Order (tag 35-MsgType=D)
- Order Cancel/Replace Request (tag 35-MsgType=G)
- New Order Cross (tag 35-MsgType=s)
- Mass Quote (tag 35-MsgType=i)

Outgoing

- Execution Report: Order Creation or Modify (tag 35-MsgType=8, OrdStatus=0 or 5)
- Execution Report: Order Cancel (tag 35-MsgType=8, OrdStatus=4)
- Execution Report: Fill Notice (tag 35-MsgType=8, tag 39-OrdStatus=1 or 2)
  - for spreads and outrights
- Execution Report: Order Elimination (tag 35-MsgType=8, tag 39-OrdStatus=4 or C)

- Execution Report: Reject (tag 35-MsgType=8, tag 39-OrdStatus=8)
- Execution Report: Trade Cancel (tag 35-MsgType=8, tag 39-OrdStatus=H)Order Cancel/Replace Reject (tag 35-MsgType=9, tag 434-CancelRejResponseTo=2)
  - only for outrights
- Quote Acknowledgement (tag 35-MsgType=b)

Market Data Security Status (tag 35-MsgType=f) Message

The Security Status (35=f) message has been enhanced to more efficiently reflect: the market state, the reason an instrument or group is entering the state, and the events that accompany the state. The Security Status (35=f) message provides a more comprehensive status.

The Security Status (35=f) message is the source of the current Trade Date (tag 75-TradeDate) on the Incremental market data feed instead of the Market Data Incremental Refresh (35=X) message.

The following changes were made to the Security Status (35=f) message:

- Differentiation between Pre-Open state and Post-Close state
- Differentiation between Opening state with new price indication from Open state
- Improved reason of status change classification (adding Expiration and Activation)
- Improved Trading Status Event classification
- Added support for Implied ON and Implied OFF notification, both related and unrelated to a state change
- Added new tag: tag 6937-Asset - contains the value previously sent in tag 1151-SecurityCode

The following three tags, in combination, identify the state, the reason for the state change, and additional events or rules that are related to the state:

- Tag 326-SecurityTradingStatus
- Tag 327-HaltReason
- Tag 1174-SecurityTradingEvent

Quote Request (tag 35-MsgType=R) Message

The following changes were made to the Quote Request (tag 35=R) message.

For the tag 146-NoRelatedSym repeating group:

- Tag 60-TransactionTime has been moved from the tag 146-NoRelatedSym repeating group to the message body.
- Tag 22-SecurityIDSource was removed from the message because it contains a constant value.

Market Data Snapshot Full Refresh (tag 35-MsgType=W) Message

A Market Data Snapshot Full Refresh (35=W) message is generated only if the instrument has had book/trade activity since the beginning of the week. Once the instrument has had at least one order, it will stay in the loop until the end of the week, even though the order may have been cancelled. Market Data Snapshot Full Refresh (35=W) messages are not sent for the instruments that have only settlements, price bands, or thresholds, but have had no book activity since the beginning of the week.

The following information is available in the Market Data Snapshot Full Refresh (35=W) message:

- Book Quotes – Bids and Offers
- Implied Book Quotes – Bids and Offers

- Last Trade
- Opening Prices
- Session High and Low Trade Prices
- Session High Bid and Session Low Offer
- Fixing Price
- Settlement Price *
- Trade Volume *
- Open Interest *
- Electronic Volume *
- Threshold Limits *

Information added to the Snapshot Full Refresh (35=W) message for the MDP 3.0 release is indicated with an asterisk (*).

The following statistics are available in the Market Data Snapshot Full Refresh (35=W) message:

- Settlement Price
- Trade Volume
- Open Interest
- Electronic Volume
- Threshold Limits

When the Group Status Message is sent with tag 1174-SecurityTradingEvent=4 (Change of Trading Session/Reset Stats), the following statistics will be cleared until new statistics are received from the Incremental Feed:

- Last Trade
- Opening Price
- Session High Trade Price
- Session Low Trade Price
- Session High Bid
- Session Low Offer
- Electronic Volume

For FIX/FAST, client systems compare the value of tag 34-MsgSeqNum to the value of tag 369-LastMsgSeqNumProcessed to determine whether any messages have been missed. For MDP 3.0, client systems should compare the packet sequence number to tag 369-LastMsgSeqNumProcessed in the Snapshot message to determine whether any messages have been missed.

In addition to sequence, tag 779-LastUpdateTime allows the client system to determine whether it must recover updates to instruments on the channel based on time of the last update.

Tag 731-SettlPriceType contains a bitmap field of Boolean type indicators representing the settlement price type as described for the Incremental Refresh message above.

Tag 286-OpenCloseSettlFlag is no longer sent for settlement prices.

Market Data Security Definition (tag 35-MsgType=d) Message

The following enhancements were made to the Market Data Security Definition (35=d) message:

- Tag 55-Symbol and tag 1151-SecurityGroup were enhanced so that tag 55-Symbol reflects the instrument attributes and tag 1151-SecurityGroup reflects the group of related securities. Tag 107-SecurityDesc is no

longer sent.

> These changes result in inconsistent data when comparing these tags to iLink Messages. Previously tag 55-Symbol in Market Data messages matched the same tag in Order Entry messages. With the implementation of MDP 3.0, these tags do not contain the same values

The table below shows the relationship between the MDP and iLink tags.

Previous MDP Format	MDP 3.0 (New Market Data Format)	iLink Order Entry
tag 107-SecurityDesc	tag 55-Symbol	tag 107-SecurityDesc
tag 55-Symbol	tag1151-SecurityGroup	tag 55-Symbol
tag1151-SecurityGroup	tag 6937-Asset	tag1151-SecurityGroup

- Along with the tag 55-Symbol and tag 1151-SecurityGroup enhancements, tag 311-UnderlyingSymbol was enhanced to reflect the new tag 55-Symbol format for the underlying instruments.
- Tag 1145-EventTime will include the date in addition to the time, in UTC format.
- Additional valid values will be added to tag 872-InstAttribValue:

  - 18 = GT Orders Eligibility (previously tag 827-ExpirationCycle)

  - 19 = Implied Matching Eligibility (previously tag 1144-ImpliedMarketIndicator)

The following new tags were added to the Market Data Security Definition (35=d) message:

- tag 779-LastUpdateTime indicates the timestamp for when the instrument was last added, modified or deleted.
- tag 1300-MarketSegmentID identifies the market segment for all CME Globex Instruments.
- tag 6937-Asset indicates the underlying asset code, which was previously disseminated in tag 1151-SecurityGroup.
- tag 167-SecurityType indicates the type of security.
- tag 6350-TickRule indicates the VTT Code provided for instruments with variable ticks in addition to the Tick Size in Tag 969-MinPriceIncrement.
- tag 37702-MainFraction indicates the price denominator of the main fraction.
- tag 37703-SubFraction indicates the price denominator of the sub fraction.
- tag 9800-PriceDisplayFormat indicates the number of decimals in the displayed price (e.g., for price '100.25' will = '2')
- tag 731-SettlPriceType is sent for settlement prices. It will contain a bitmap field of Boolean type indicators representing the settlement price type as described for the Incremental Refresh message.

> Tag 286-OpenCloseSettlFlag is no longer sent for settlement prices.

The following tags were removed from the Market Data Security Definition (35=d) message:

- tag 5796-TradingReferenceDate **Note:** this tag will be be re-introduced at a future date TBD.
- tag 827-ExpirationCycle
- tag 107-SecurityDesc
- tag 9853-PricingModel
- tag 454-NoSecurityAltID with repeating groups: tag 455-SecurityAltID and tag 456-SecurityAltIDSource
- tag 1144-ImpliedMarketIndicator (implied eligibility for instruments will be found in the tag 871-InstAttribType=24 repeating group).
- tags from the Leg repeating group (tag 555-NoLegs):

  - tag 5795-LegSecurityGroup

- tag 600-LegSymbol
- tag 608-LegCFICode
- tag 764-LegSecuritySubType
- tag 556-LegCurrency
- tag 610-LegMaturityMonthYear
- tag 612-LegStrikePrice
- tag 616-LegSecurityExchange
- tag 620-LegSecurityDesc
- tag 942-LegStrikeCurrency
- tag 286-OpenCloseSettlFlag

Unchanged Messages

There were no changes to the following market data messages:

- Market Data Heartbeat (tag 35-MsgType=0)
- Market Data Logon (tag 35-MsgType=A)
- Market Data Logoff (tag 35-MsgType=5)
- Market Data Replay Request (tag 35-MsgType=V)

Recovery

The Recovery Feeds were not changed. Client systems will request messages to be replayed using the Packet Sequence Number from the Packet Header instead of the Message Sequence Number (tag 34-MsgSeqNum).

An additional location was added to the CME ftp site (ftp://ftp.cmegroup.com/SBEFix) for message schemas and config.xml files for the MDP 3.0 Platform.

Book Management was not changed. Client systems should process the Incremental Refresh book updates as they currently process them.

Channel Reset

The Channel Reset process includes the following enhancements:

- The Channel Reset message includes the Channel ID (tag 1180-ApplID) for the channel that is being reset.
- Individual Order Book Reset data blocks for each instrument are no longer sent in addition to the Channel Reset data block.
- All messages resent during recovery will contain Bit 5 set to '1' in tag 5799-MatchEventIndicator.
- Additional statistics data are recovered (e.g., Electronic Volume, Indicative Opening Price, Trade High Price, Trade Low Price), where applicable, by instrument.

See MDP3 - Channel Reset for further information.

Order Routing Partner Exchanges

The schedule for implementation of the MDP 3.0 format for the Order Routing Partner Exchanges, BM&FBOVESPA and MexDer (Mercado Mexicano de Derivados), is as follows:

Environment	BVMF	MexDer
New Release	July 2015	August 2015
AutoCert+ Test Suite Available	August 2015	September 2015
Production Parallel	August 2015	September 2015

AutoCert+ certification of the MDP 3.0 and order routing partner exchange (BVMF and MexDer) test suites is mandatory prior to receiving Market Data for each order routing partner exchange.

Streamlined FIX/FAST

The Streamlined FIX/FAST Market Data Channels will not be upgraded to FIX Binary and SBE until after MDP 3.0 is implemented. The implementation schedule will be announced at a later date.